CH32V00X Reference Manual

V1.4

https://wch-ic.com

Overview

CH32V00X series are industrial-grade general-purpose microcontrollers designed based on 32-bit RISC-V instruction set and architecture. It adopts QingKe V2C core, RV32EC instruction set, and supports 2 levels of interrupt nesting. The series are mounted with rich peripheral interfaces and function modules. Its internal organizational structure meets the low-cost and low-power embedded application scenarios.

This manual provides detailed information on the use of the CH32V00X series for the user's application development, and is applicable to products with different memory capacities, functional resources, and packages in the series; any differences will be specially explained in the corresponding functional chapters.

Refer to the following datasheets for device characteristics of this series.

CH32V002: *CH32V002DS0*

CH32V004: *CH32V004DS0*

CH32V005, CH32V006: *CH32V006DS0*

CH32V007, CH32M007: *CH32V007DS0*

For information about the core, refer to the QingKeV2 Microprocessor Manual, QingKeV2_Processor_Manual.

RISC-V core version overview

Features / Core versions	Instruction set	Hardware stack levels	Interrupt nesting levels	Fast interrupt channels	Flow line	Vector table model	Extensions instruction	Debug interface
QingKe V2C	RV32EmC	2	2	2	2	Address or command	Support	1-wire

Note: The "m" extension in RV32EmC implements the multiplication subset of the M extension.

Abbreviated description of the bit attribute in the register:

Register bit properties	Property description
RF	Read-only property that reads a fixed value.
RO	Read-only attribute, changed by hardware.
RZ	Read-only property, auto bit clear 0 after read operation.
WO	Write-only attribute (not readable, read value uncertain)
WA	Write-only attribute, writable in Safe mode.
WZ	Write-only attribute, auto bit clear 0 after write operation.
RW	Readable and writable.
RWA	Readable, writable in Safe mode.
RW1	Readable, write 1 is valid, write 0 is invalid.
RW0	Readable, write 0 valid, write 1 invalid.

RW1T	Readable, write 0 invalid, write 1 flipped.

Chapter 1 Memory and Bus Architecture

1.1 Bus Architecture

CH32V00X series products are designed based on RISC-V instruction set, and its architecture realizes the interaction of core, arbitration unit, DMA module, SRAM memory and other parts through multiple buses. The design integrates a general-purpose DMA controller to reduce the burden on the CPU and improve the access efficiency, while at the same time, it also has a data protection mechanism, automatic clock switching protection, and other measures to increase the stability of the system.

Figure 1-1 CH32V002 system block diagram

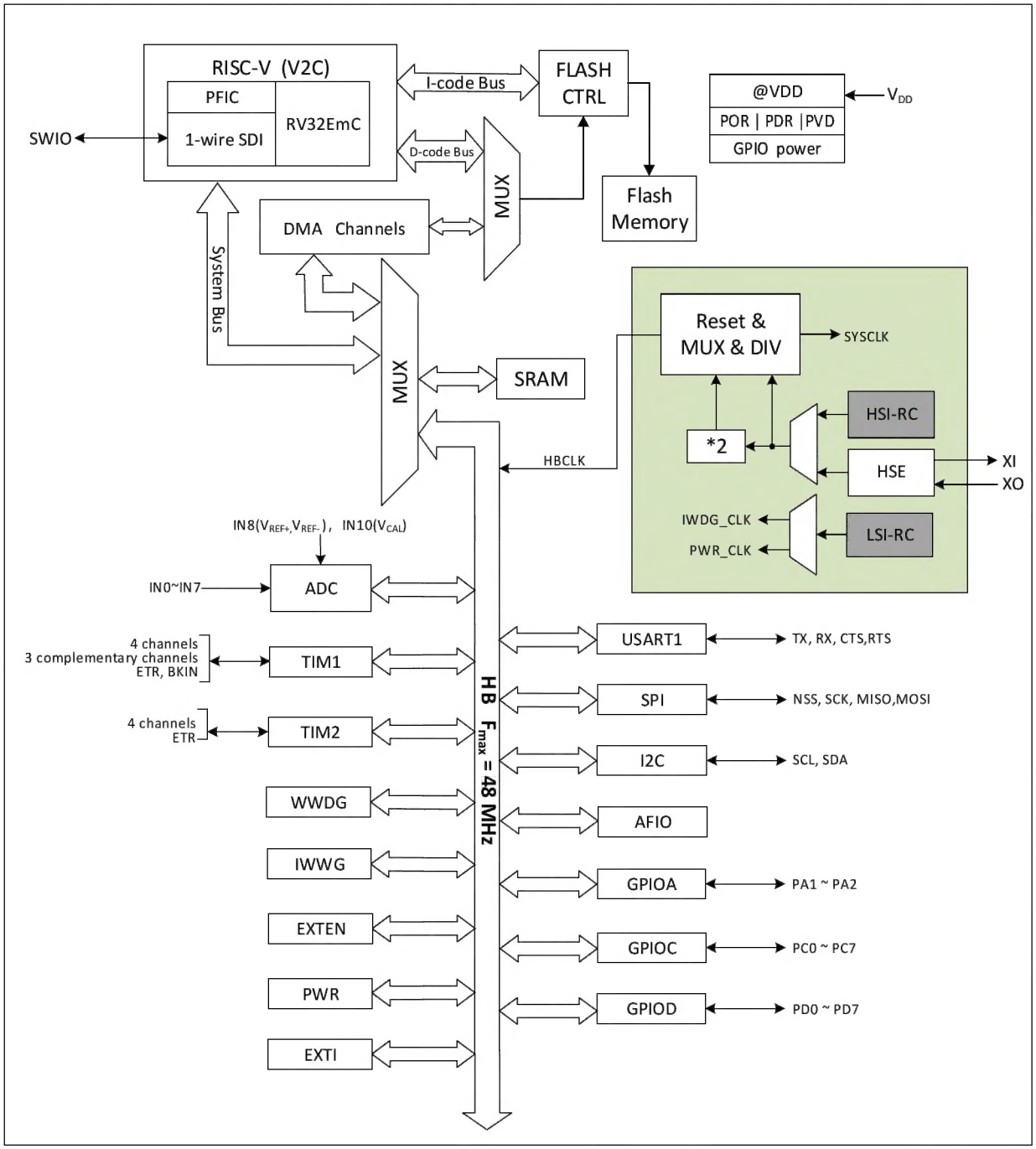


Figure 1-2 CH32V004 system block diagram

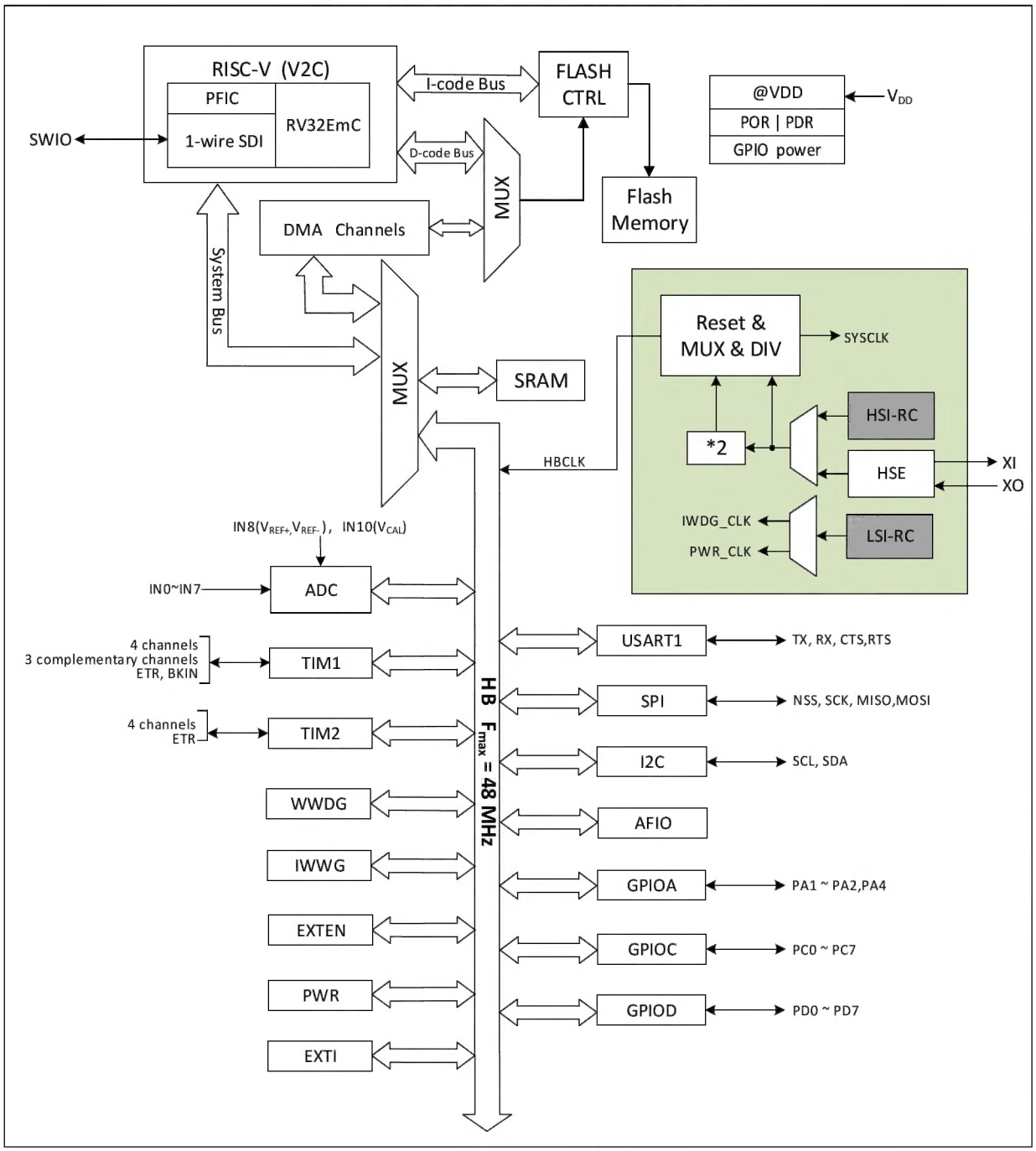


Figure 1-3 CH32V005, CH32V006 system block diagram

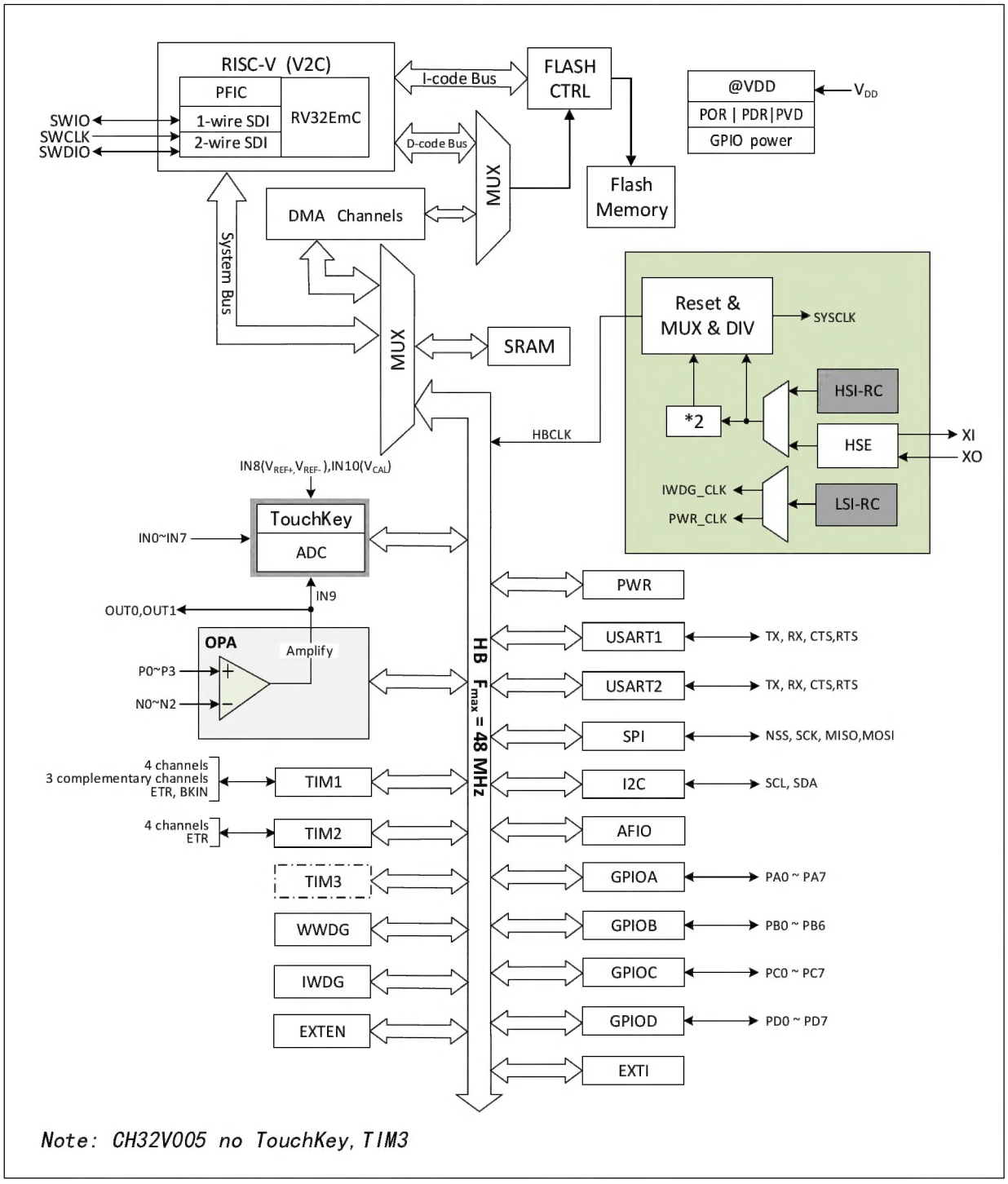


Figure 1-4 CH32V007 system block diagram

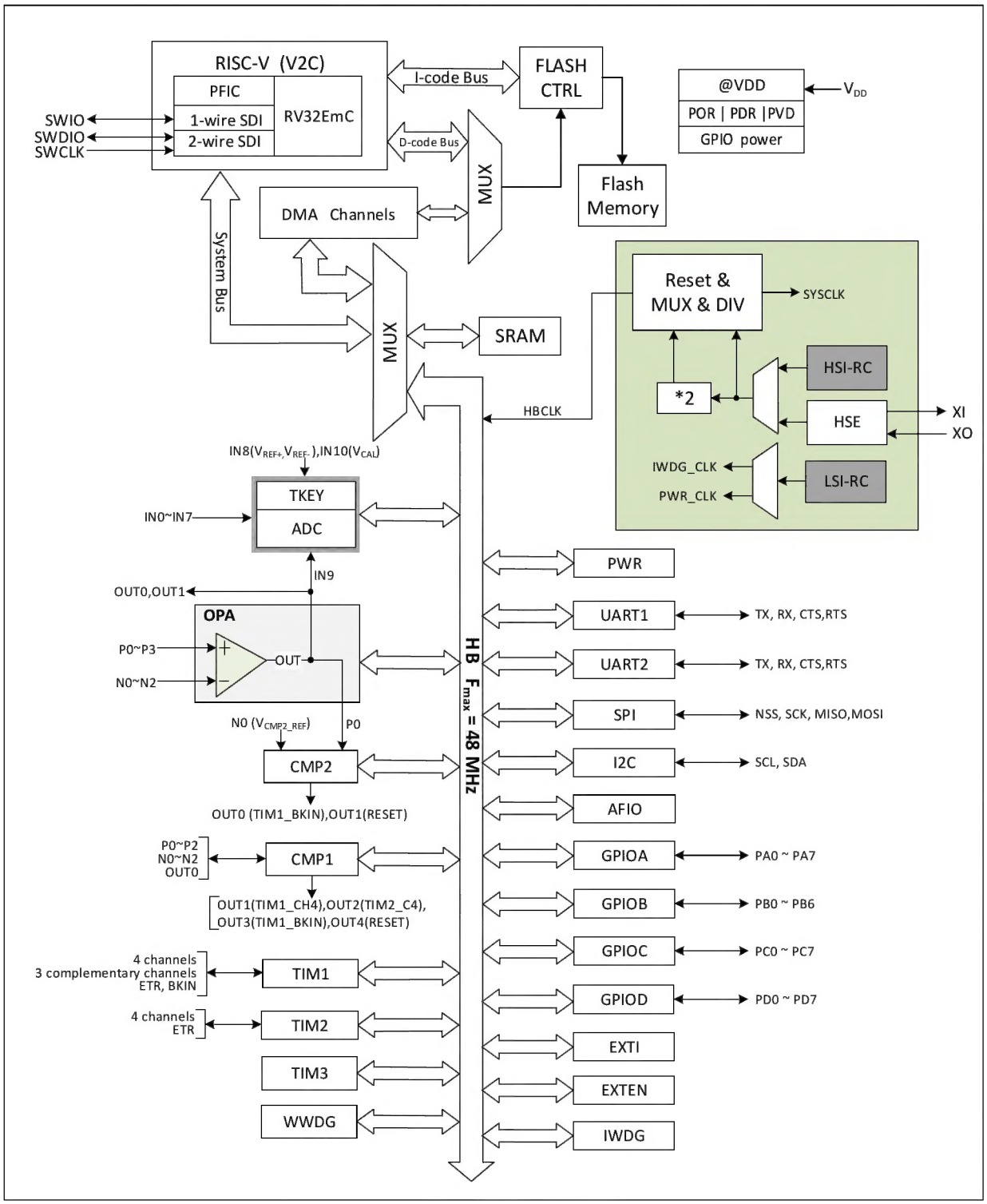


Figure 1-5-1 CH32M007G8R6 system block diagram

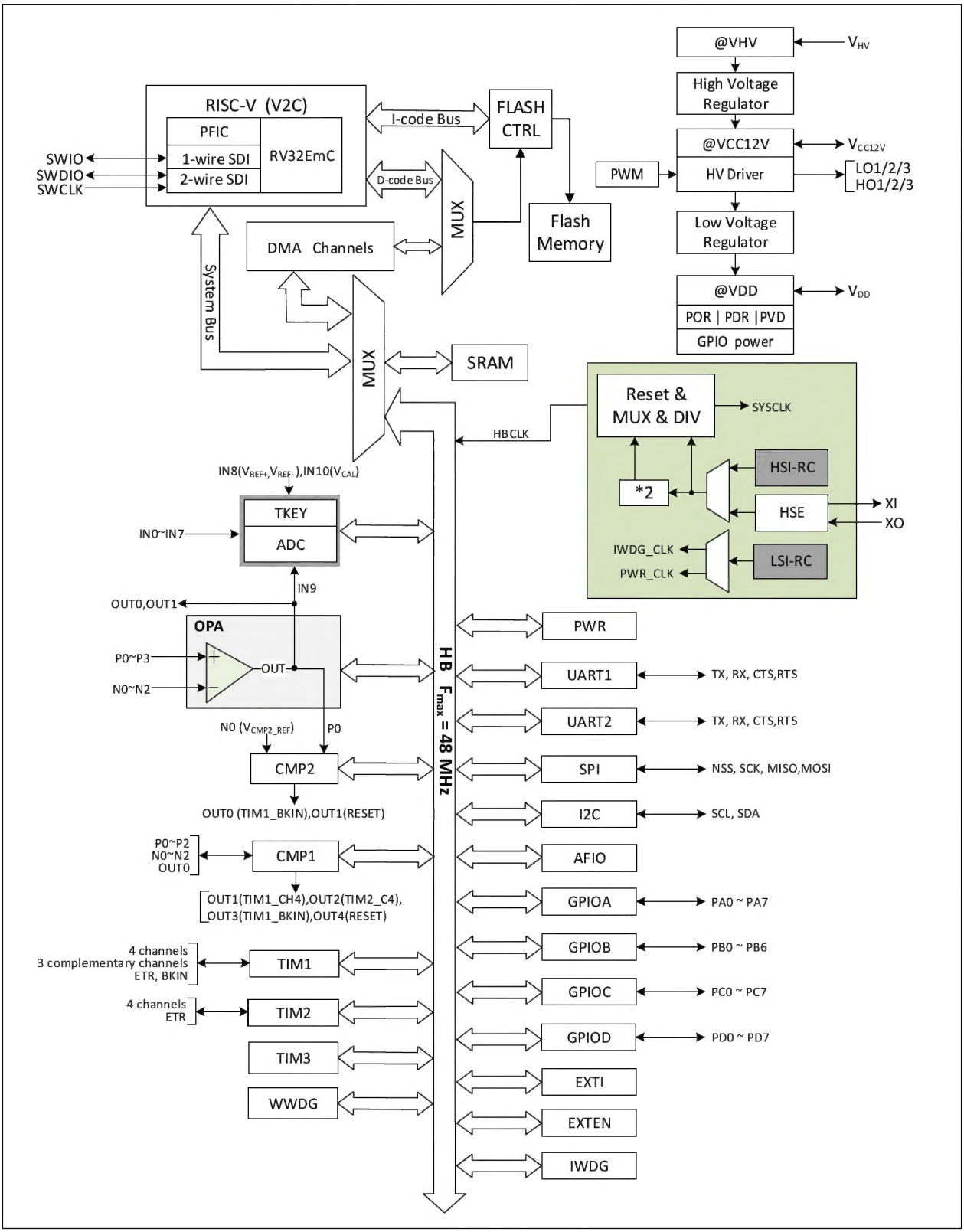


Figure 1-5-2 CH32M007E8U6 system block diagram

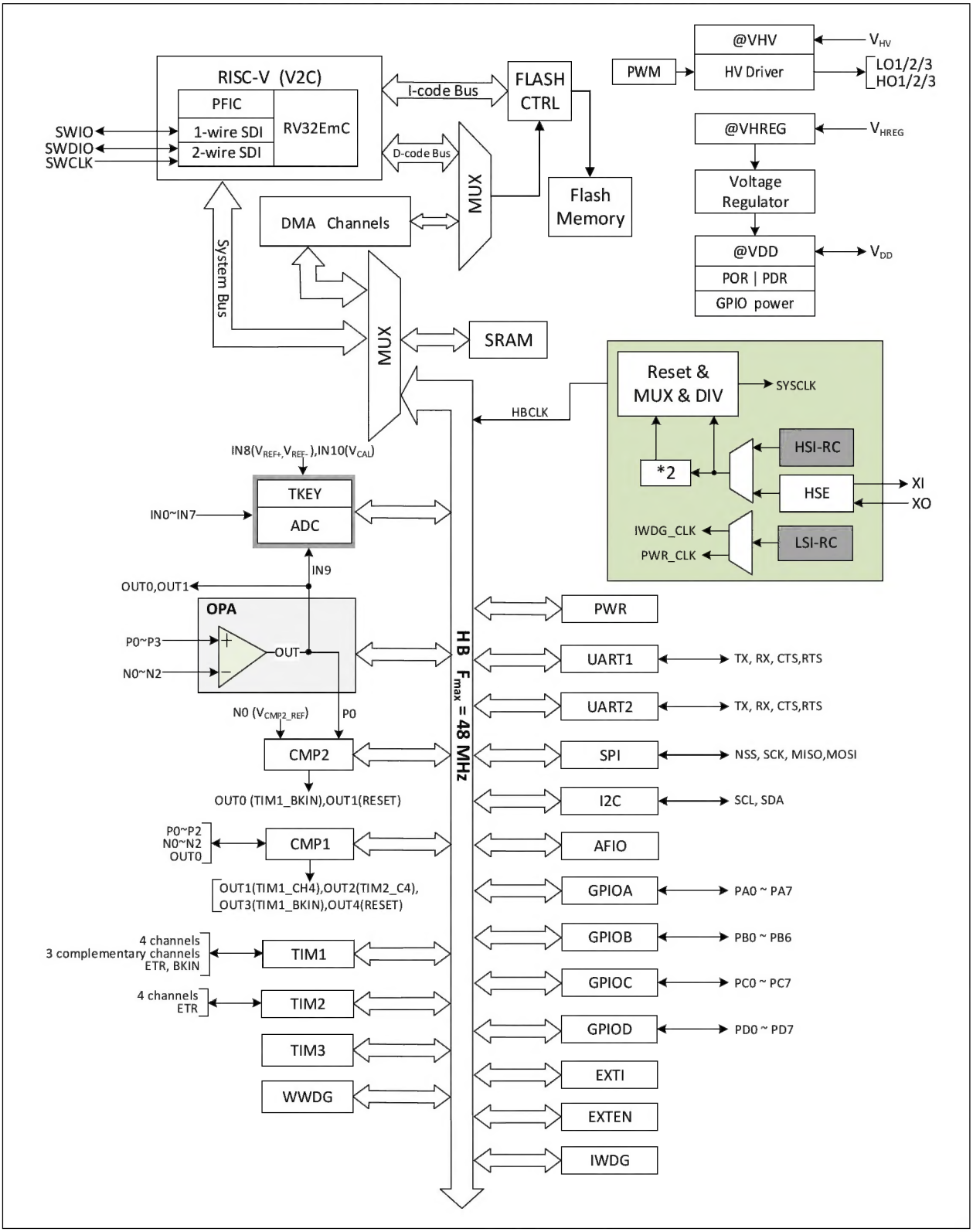
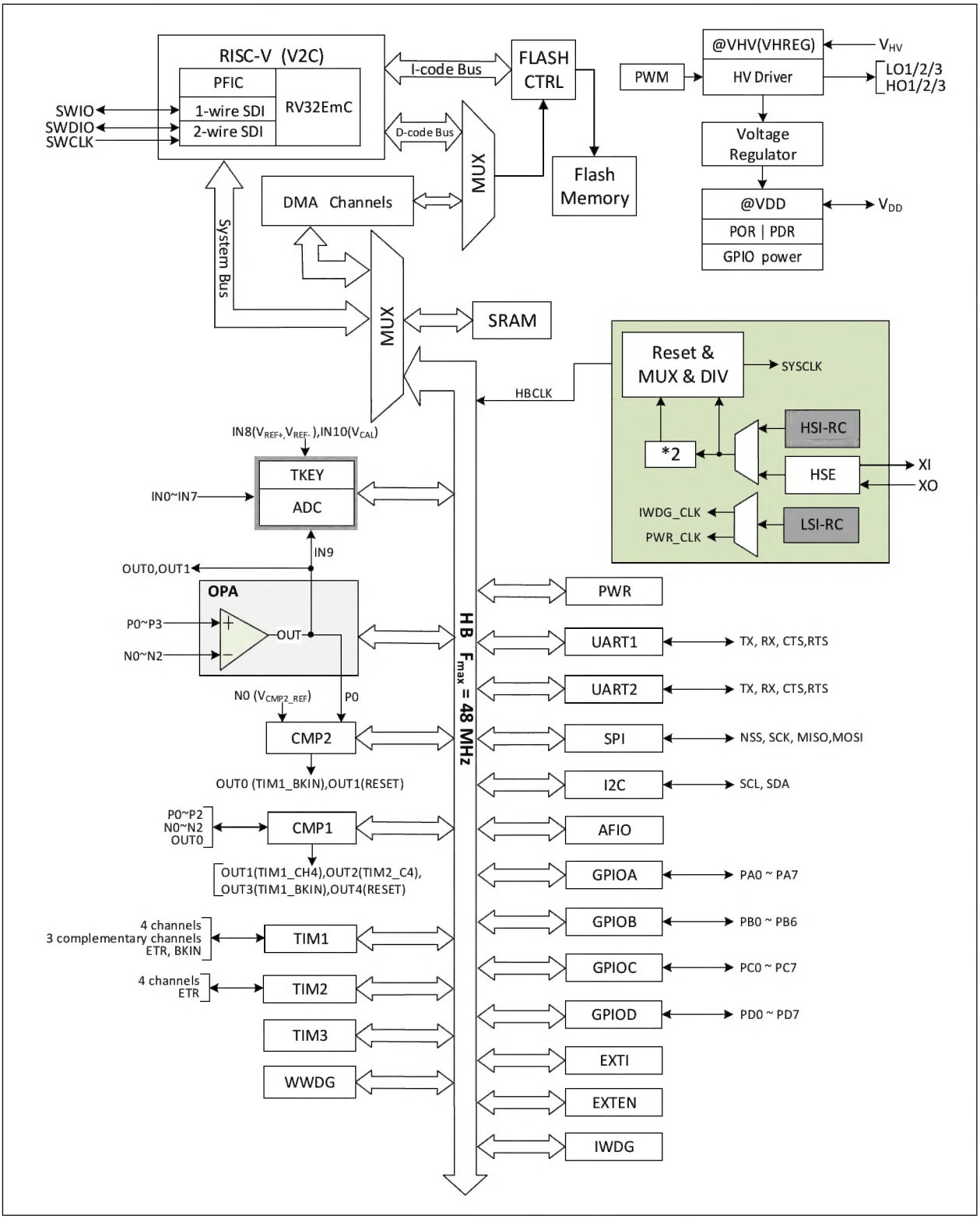


Figure 1-5-3 CH32M007E8R6 system block diagram



The system is equipped with a general DMA controller to reduce the burden of CPU and improve efficiency, and clock tree hierarchical management to reduce the total power consumption of peripherals, as well as data protection mechanism, clock security system protection mechanism and other measures to increase system stability.

- The instruction bus (I-Code) connects the core to the FLASH instruction interface and prefetching is done on this bus.
- The data bus (D-Code) connects the core to the FLASH data interface for constant loading and debugging.

- The system bus connects the core to the bus matrix and is used to coordinate accesses to the core, DMA, SRAM and peripherals.
- The DMA bus is responsible for the DMA of the HB master interface connected to the bus matrix, which is accessed by FLASH data, SRAM and peripherals.
- The bus matrix is responsible for the access coordination between the system bus, data bus, DMA bus, SRAM and HB bridge.

1.2 Memory Image

The CH32V00X family contains program memory, data memory, core registers, peripheral registers, and more, all addressed in a 4GB linear space.

System storage stores data in small-end format, i.e., low bytes are stored at the low address and high bytes are stored at the high address.

Figure 1-6 CH32V002 Storage image

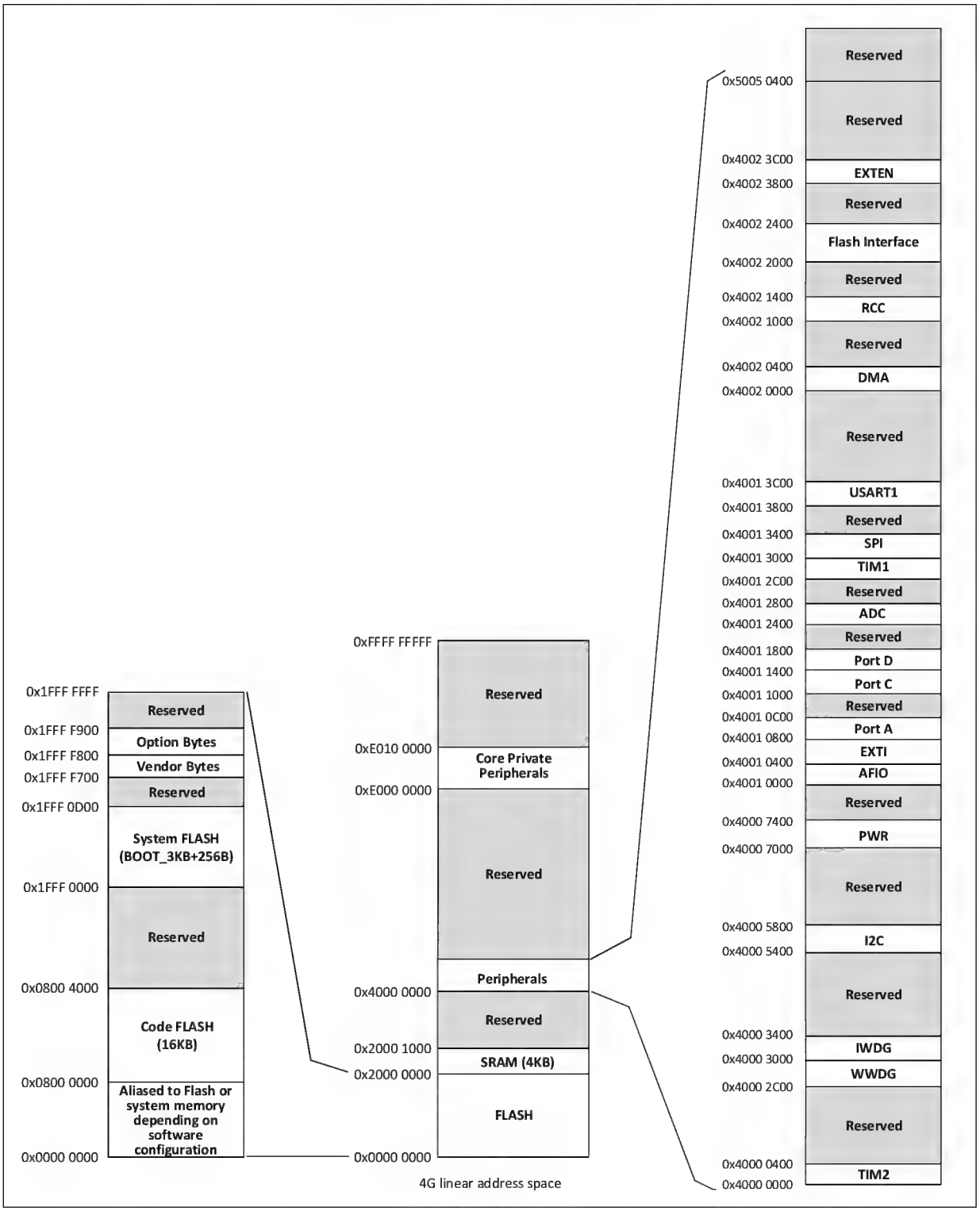


Figure 1-7 CH32V004 Storage image

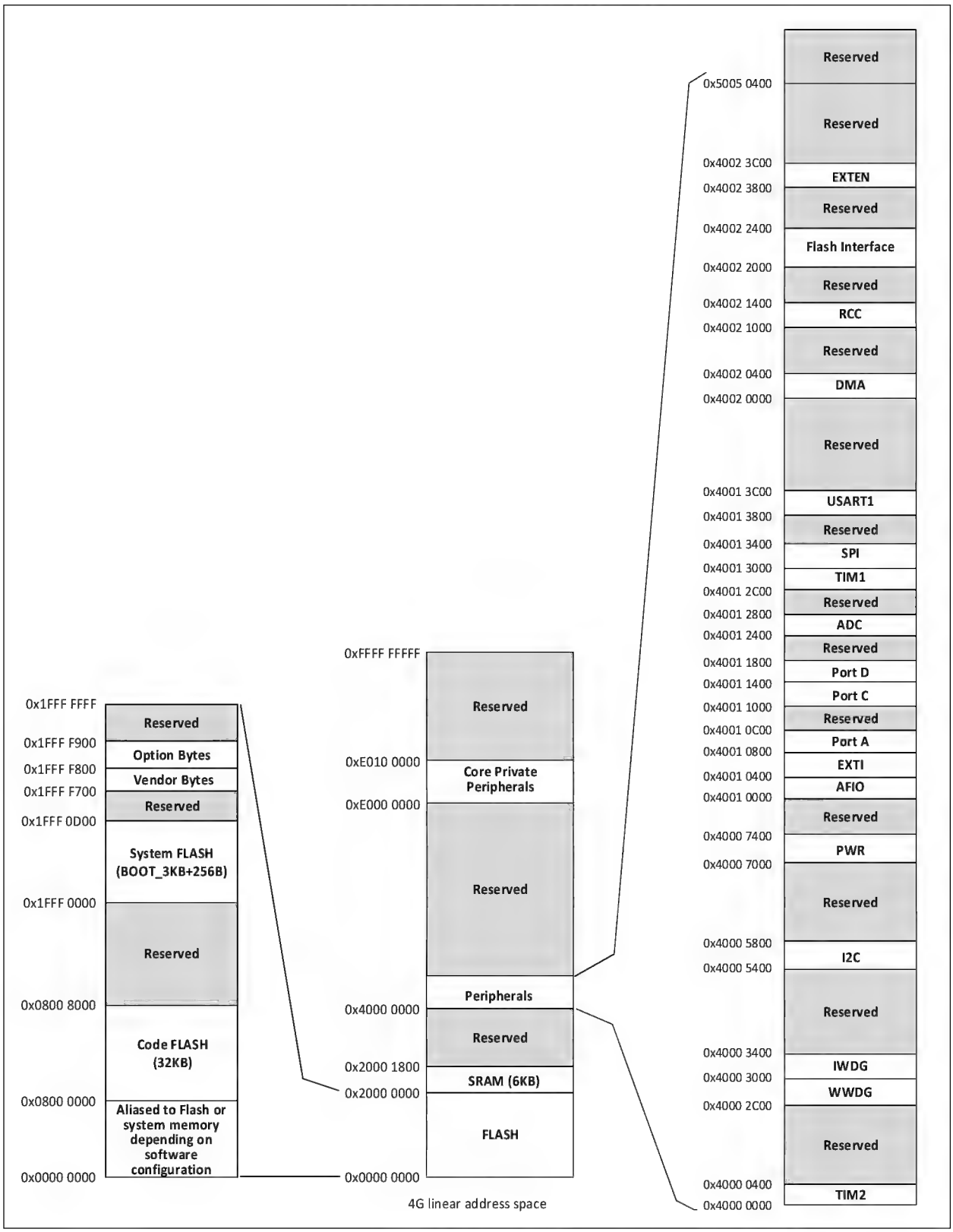


Figure 1-8 CH32V005, CH32V006 Storage image

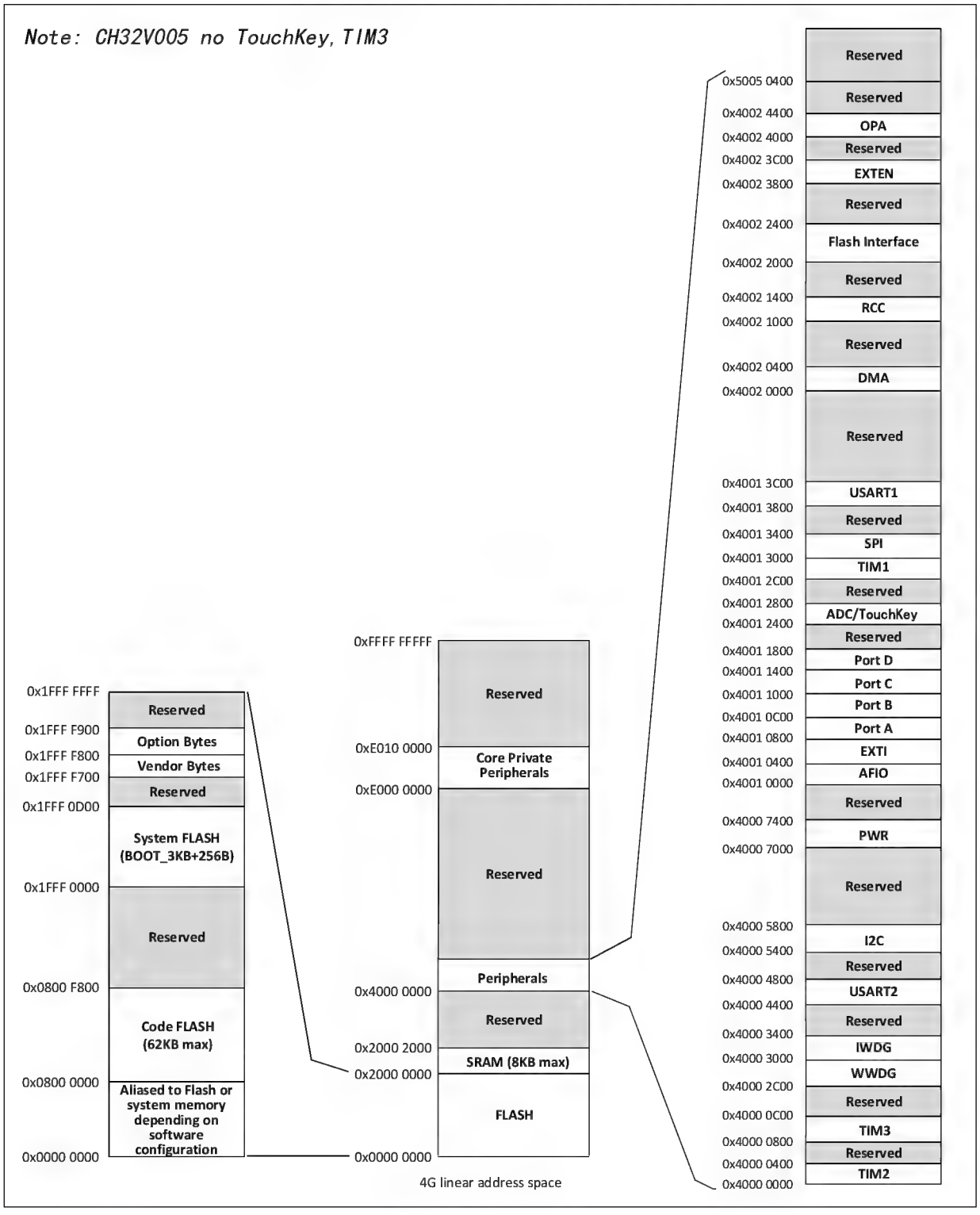
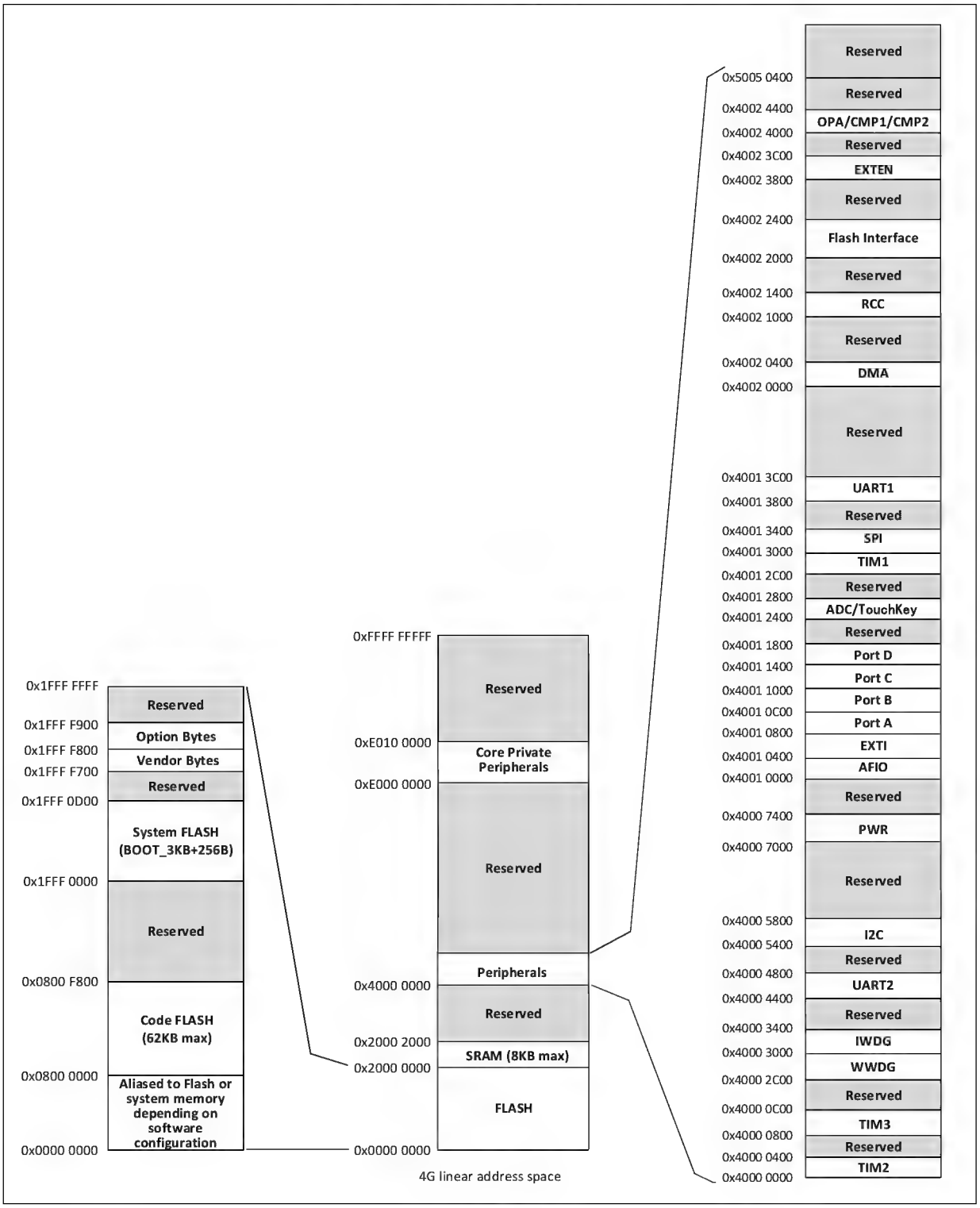


Figure 1-9 CH32V007, CH32M007 Storage image



1.2.1 Memory Allocation

SRAM start address 0x20000000, supports byte, half-word (2-byte), and full-word (4-byte) accesses for storing data, which is lost after power-down.

Program flash memory storage area (Code FLASH), i.e. user area, is used for user's application program and constant data storage.

System storage area (System FLASH), i.e. BOOT area, is used for system boot program storage (Manufacturer's

cured bootloader).

User-defined information storage area for user option byte storage.

Note: Please refer to the corresponding datasheet for the details of the storage allocation for each model.

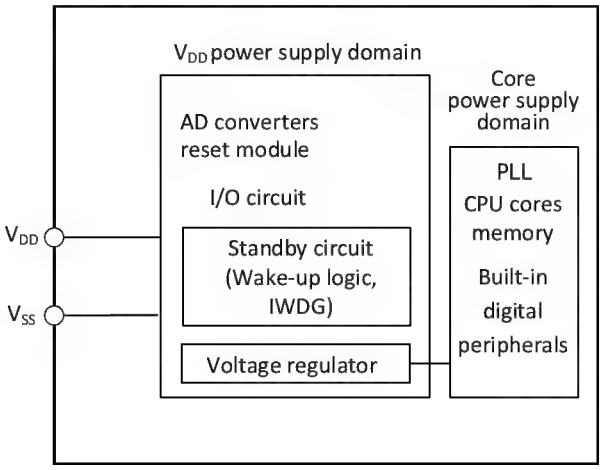
Chapter 2 Power Control (PWR)

The module descriptions in this chapter apply to the full range of CH32V00X microcontrollers.

2.1 Overview

For CH32V002, CH32V004, CH32V005, CH32V006, CH32V007, the operating voltage of the system ranges from 1.9V to 5.5V, and the built-in voltage regulator provides the working power required by the core. The power supply structure is shown in figure 2-1.

Figure 2-1 Block diagram of power supply structure



For CH32M007:

V_{HV}: Supplies power to the internal high voltage regulator.

V_{CC12V}: Supplies power to the internal gate driver as well as the internal low voltage regulator.

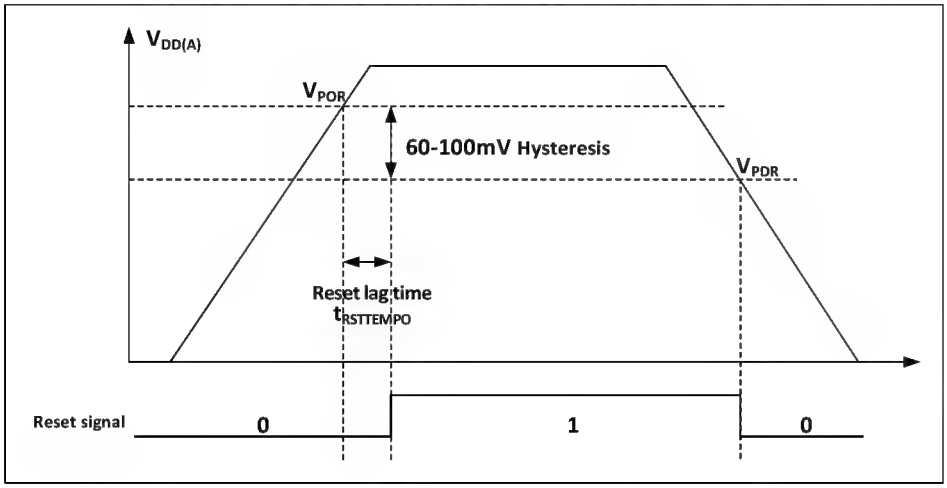
V_{DD}: Supplies power to the I/O pins.

2.2 Power Management

2.2.1 Power-on Reset and Power-down Reset

The power-on reset POR and power-off reset PDR circuits are integrated in the system. When the chip supply voltage V_{DD} is lower than the corresponding threshold voltage, the system is reset by the relevant circuits, and there is no need for additional external reset circuits. For the parameters of power-on threshold voltage V_{POR} and power-off threshold voltage V_{PDR}, please refer to the corresponding data manual.

Figure 2-2 Schematic diagram of the operation of POR and PDR



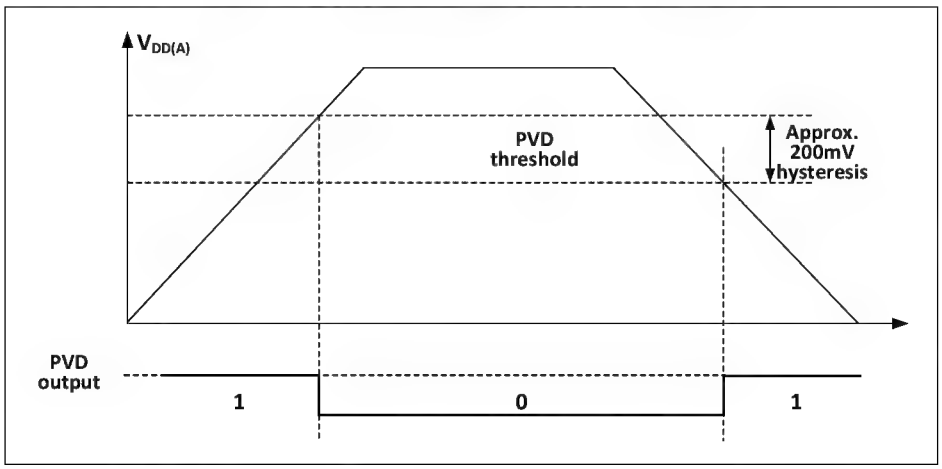
2.2.2 Programmable Voltage Detector

The programmable voltage monitor, PVD, is mainly used to monitor the change of the main power supply of the system and compare it with the threshold voltage set by PLS[1:0] of the power control register PWR_CTLR, and with the external interrupt register (EXTI) setting, it can generate relevant interrupts to notify the system in time for pre-power down operations such as data saving.

The specific configuration is as follows.

1) Set the PLS[1:0] field of the PWR_CTLR register to select the voltage threshold to be monitored.
2) Optional interrupt handling. the PVD function internally connects to the rising/falling edge trigger setting of line 8 of the EXTI module, turns on this interrupt (Configures EXTI), and generates a PVD interrupt when V_{DD} drops below the PVD threshold or rises above the PVD threshold.
3) Set the PVDE bit of PWR_CTLR register to enable the PVD function.
4) Read the PVD0 bit of PWR_CSR status register to obtain the current system main power and PLS[1:0] setting threshold relationship, and perform the corresponding soft processing. When the V_{DD} voltage is higher than the threshold set by PLS[1:0], PVD0 position 0; when the V_{DD} voltage is lower than the threshold set by PLS[1:0], PVD0 position 1.

Figure 2-3 Schematic diagram of PVD operation



Note: The function is apply for CH32V002, CH32V005, CH32V006, CH32V007, CH32M007 model chips.

2.3 Low-power Modes

After a system reset, the microcontroller is in a normal operating state (Run mode), where system power can be saved by reducing the system main frequency or turning off the unused peripheral clock or reducing the operating peripheral clock. If the system does not need to work, you can set the system to enter low-power mode and let the system jump out of this state by specific events.

Microcontrollers currently offer 2 low-power modes, divided in terms of operating differences between processors, peripherals, voltage regulators, etc.

- Sleep mode: The core stops running and all peripherals (Including core private peripherals) are still running.
- Standby mode: Stop all clocks, wake up and switch the clock to HSI.

Table 2-1 Low-power mode list

Mode	Enter	Wakeup source	Effect on clock	Voltage regulator
Sleep	WFI	Any interrupt	Core clock OFF, no effect on other clocks	Normal
	WFE	Wake-up event		
Standby	Set SLEEPDEEP to 1 Set PDDS to 1 WFI or WFE	Any interrupt/event (EXTI signal), AWU event, PVD output, NRST pin reset, IWDG reset. *Note: Any events can wake up the system, but the system is not reset after wakeup.*	HSE, HSI, PLL and peripheral clock OFF	Low-power mode

Note: SLEEPDEEP bit belongs to core private peripheral control bit, CH32V00X series products refer to PFIC_SCTLR register.

2.3.1 Low-power Configuration Options

- WFI and WFE

WFI: The microcontroller is woken up by an interrupt source with interrupt controller response, and the interrupt service function will be executed first after the system wakes up (Except for microcontroller reset).

WFE: The wakeup event triggers the microcontroller to exit low-power mode. Wake-up events include.

1) Configure an external or internal EXTI line to event mode, when no interrupt controller needs to be configured.
2) Or configure an interrupt source, equivalent to a WFI wakeup, where the system prioritizes the execution of the interrupt service function.
3) Or configure the SEVONPEND bit to turn on peripheral interrupt enable, but not interrupt enable in the interrupt controller, and the interrupt pending bit needs to be cleared after the system wakes up.

- SLEEPONEXIT

  Enable: After executing the WFI or WFE instruction, the microcontroller ensures that all pending interrupt services are exited and then enters low-power mode.

  Not enabled: The microcontroller enters low-power mode immediately after executing the WFI or WFE command.

- SEVONPEND

  Enable: All interrupts or wake-up events can wake up the low-power consumption entered by executing WFE.

  Not enabled: Only interrupts or wake-up events enabled in the interrupt controller can wake up the low-power consumption entered by executing WFE.

2.3.2 Sleep Mode (SLEEP)

In this mode, all I/O pins keep their state in Run mode and all peripheral clocks are normal, so try to turn off useless peripheral clocks before entering Sleep mode to reduce low-power consumption. This mode takes the shortest time to wake up.

Enter: Configure core register control bit SLEEPDEEP=0, power control register PDDS=0, execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit: Arbitrary interrupt or wakeup event.

2.3.3 Standby Mode (STANDBY)

Standby mode is a combination of peripheral clock control mechanisms based on the core's deep Sleep mode (SLEEPDEEP) and allows the voltage regulator to operate at a much lower-power consumption. This mode has the high frequency clock (HSE/HSI/PLL) domain turned off, the SRAM and register contents held, and the I/O pin state held. The system can continue to run after this mode wakes up, and the HSI is called the default system clock.

If flash programming is in progress, the system does not enter Standby mode until access to memory is complete.

Standby mode can work modules: Independent Watchdog (IWDG), Low Frequency Clock (LSI).

Enter: Configure the core register control bit SLEEPDEEP=1, PDDS=1 in the power control register, and execute WFI or WFE, optionally SEVONPEND and SLEEPONEXIT.

Exit: Any external interrupt/event (EXTI signal), external reset signal on RST, PVD output, IWDG reset, AWU automatic wake-up, etc.

2.3.4 Auto-wakeup (AWU)

It can wake up automatically without external interruption. By programming the time base, it can be awakened periodically from standby mode.

When you turn on the AWU interrupt function, you need to trigger the rising / falling edge of line 9 of the internal connection to the EXTI module to turn on this interrupt (Configure EXTI).

The optional internal low frequency 128kHz clock oscillator LSI is used as the automatic wake-up count time base.

2.4 Register Description

Table 2-2 PWR-related registers list

Name	Access address	Description	Reset value
R32_PWR_CTLR	0x40007000	Power Control Register	0x00000408
R32_PWR_CSR	0x40007004	Power Control/Status Register	0x00000000
R32_PWR_AWUCSR	0x40007008	Auto-wakeup Control/Status Register	0x00000000
R32_PWR_AWUWR	0x4000700C	Auto-wakeup Window Comparison Value Register	0x0000003F
R32_PWR_AWUPSC	0x40007010	Auto-wakeup Prescaler Factor Register	0x00000000

2.4.1 Power Control Register (PWR_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				FLASH_LP[1:0]		FLASH_LP_RE	Reserved		PLS[1:0]		PVDE	LDO_MODE[1:0]		PDDS	Reserved

		G						

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:10]	FLASH_LP[1:0]	RW	Software configuration FLASH status: 00: Idle; x1: Sleep.	01b
9	FLASH_LP_REG	RW	In conjunction with the FLASH_LP field, software configures the enable of the FLASH to enter low power mode: 1: FLASH can be put into low-power mode; 0: FLASH cannot be put into low-power mode by software.	0
[8:7]	Reserved	RO	Reserved	0
[6:5]	PLS[1:0]	RW	PVD voltage monitoring threshold setting. See the Electrical Characteristics section of the datasheet for detailed instructions. 00: Rising edge 1.87V/falling edge 1.85V; 01: Rising edge 2.23V/falling edge 2.21V; 10: Rising edge 2.43V/falling edge 2.41V; 11: Rising edge 2.66V/ falling edge 2.60V. *Note: This bit is only applicable to CH32V002, CH32V005, CH32V006, CH32V007, CH32M007 model chips.*	0
4	PVDE	RW	Power supply voltage monitoring function enable flag bit 1: Enable the power supply voltage monitoring function. 0: Disable the power supply voltage monitoring function.	0
[3:2]	LDO_MODE[1:0]	RW	Voltage regulator operating mode: 00: Unable to configure; 01: Low-power mode: LDO output 1.0V; 10: Normal mode: LDO output 1.2V; 11: Energy saving mode: LDO output 1.0V. *Note: (1) In STANDBY mode, LDO low power mode hardware is automatically turned on.* *(2) It is not recommended to run programs in flash in energy saving mode.*	10b
1	PDDS	RW	Standby/sleep mode selection bit under power-down deep sleep scenario: 1: Enter standby mode; 0: Enter sleep mode.	0
0	Reserved	RO	Reserved	0

2.4.2 Power Control/Status Register (PWR_CSR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													PVD0	Reserved	

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	RO	Reserved	0
2	PVD0	RO	PVD output status flag bits. This bit is valid when the PVDE=1 of the PWR_CTLR register. 1: VDD and VDDA are below the PVD threshold set by PLS[1:0]. 0: VDD and VDDA are above the PVD threshold set by PLS[1:0].	0
[1:0]	Reserved	RO	Reserved	0

2.4.3 Auto-wakeup Control/Status Register (PWR_AWUCSR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														AWUEN	Reserved

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
1	AWUEN	RW	Automatic wake-up enable. 1: Turn on auto-wakeup; 0: Invalid.	0
0	Reserved	RO	Reserved	0

2.4.4 Auto-wakeup Window Comparison Value Register (PWR_AWUWR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										AWUWR[5:0]					

Bit	Name	Access	Description	Reset value
[31:6]	Reserved	RO	Reserved	0

[5:0]	AWUWR[5:0]	RW	AWU window value: The AWU window value is equal to the input value of the AWU window value + 1; The AWU window value is used to compare with the up counter value. When the counter value is equal to the window value, a wake-up signal is generated.	0x3F

2.4.5 Auto-wakeup Prescaler Factor Register (PWR_AWUPSC)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												AWUPSC[3:0]			

Bit	Name	Access	Description	Reset value
[31:4]	Reserved	RO	Reserved	0
[3:0]	AWUPSC[3:0]	RW	Counting time base 0000: Prescaler off. 0001: Prescaler off. 0010: Divided by 2. 0011: Divided by 4. 0100: Divided by 8. 0101: Divided by 16. 0110: Divided by 32. 0111: Divided by 64. 1000: Divided by 128. 1001: Divided by 256. 1010: Divided by 512. 1011: Divided by 1024. 1100: Divided by 2048. 1101: Divided by 4096. 1110: Divided by 10240. 1111: Divided by 61440.	0

Chapter 3 Reset and Clock Control (RCC)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

According to the division of power supply area and the consideration of peripheral power management in application, the controller provides different reset forms and configurable clock tree structure. This chapter describes the scope of each clock in the system.

3.1 Main Features

- Multiple reset forms
- Multiple clock sources, bus clock management
- Built-in external crystal oscillation monitoring and clock security system
- Independent management of each peripheral clock: Reset, on, off
- Support internal clock output

3.2 Reset

The controller provides 2 forms of reset: power Reset and system Reset.

3.2.1 Power Reset

When a power Reset occurs, it will reset all registers.

A power Reset is generated when the following event occurs:

- Power-up/power-down reset (POR/PDR)

3.2.2 System Reset

When a system Reset occurs, it will reset the reset flag in addition to the control/status register RCC_RSTSCKR and all the registers. The source of the reset event is identified by looking at the reset status flag bit in the RCC_RSTSCKR register.

A system Reset is generated when one of the following events occurs:

- Low signal on NRST pin (External reset)
- Window watchdog count termination (WWDG reset)
- Independent watchdog count termination (IWDG reset)
- Software reset (SW reset)
- Low-power management reset
- ADC reset
- OPCM reset

Window/Independent Watchdog Reset: Generated by the window/independent watchdog peripheral timer count cycle overflow trigger, see its corresponding section for detailed description.

Software reset: The CH32V00X product resets the system via the RSTSYS position 1 of the interrupt configuration register PFIC_CFGR in the programmable interrupt controller PFIC or the SYSRST position 1 of the configuration register PFIC_SCTLR to reset the system cabinet, refer to the corresponding chapter for details.

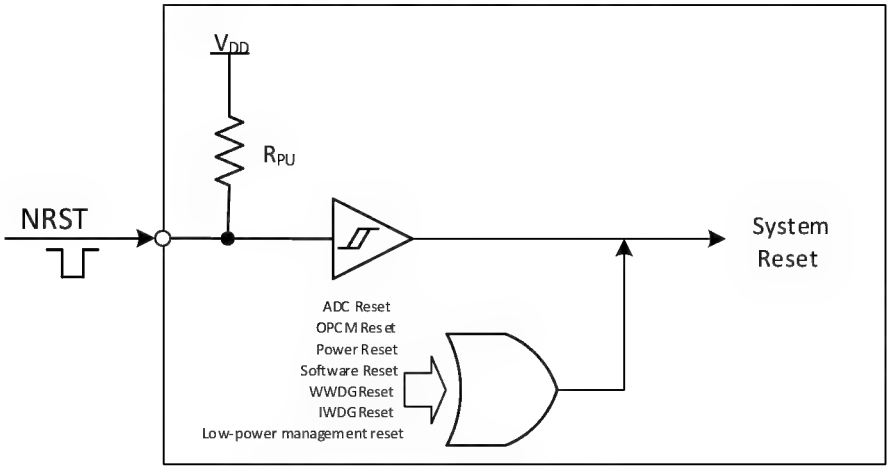
Low Power Management Reset: Standby mode reset will be enabled by setting the STANDBY_RST position 1 in the user option byte. This will perform a system reset instead of entering standby mode after the process of entering

standby mode is executed.

ADC Reset: With ADC watchdog reset enable on, an ADC reset is generated when the ADC data is greater than the watchdog high threshold or less than the watchdog low threshold.

OPCM Reset: With OPA Reset Enable and CMP Reset Enable turned on, an OPCM reset is generated when the OPA or CMP output goes high.

Figure 3-1 System reset structure



3.3 Clock

3.3.1 System Clock Structure

Figure 3-2 CH32V002, CH32V004 clock tree block diagram

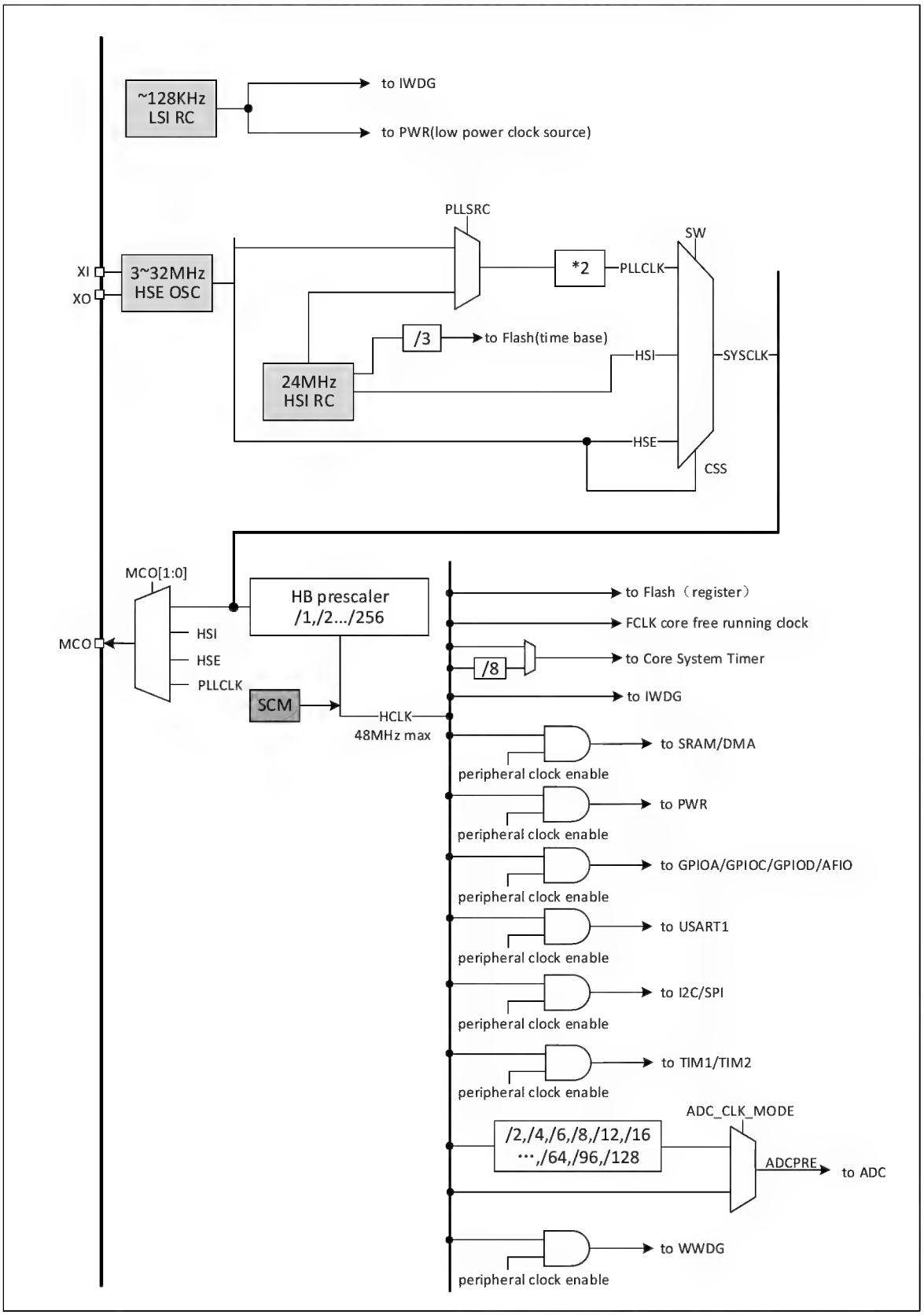


Figure 3-3 CH32V005, CH32V006 clock tree block diagram

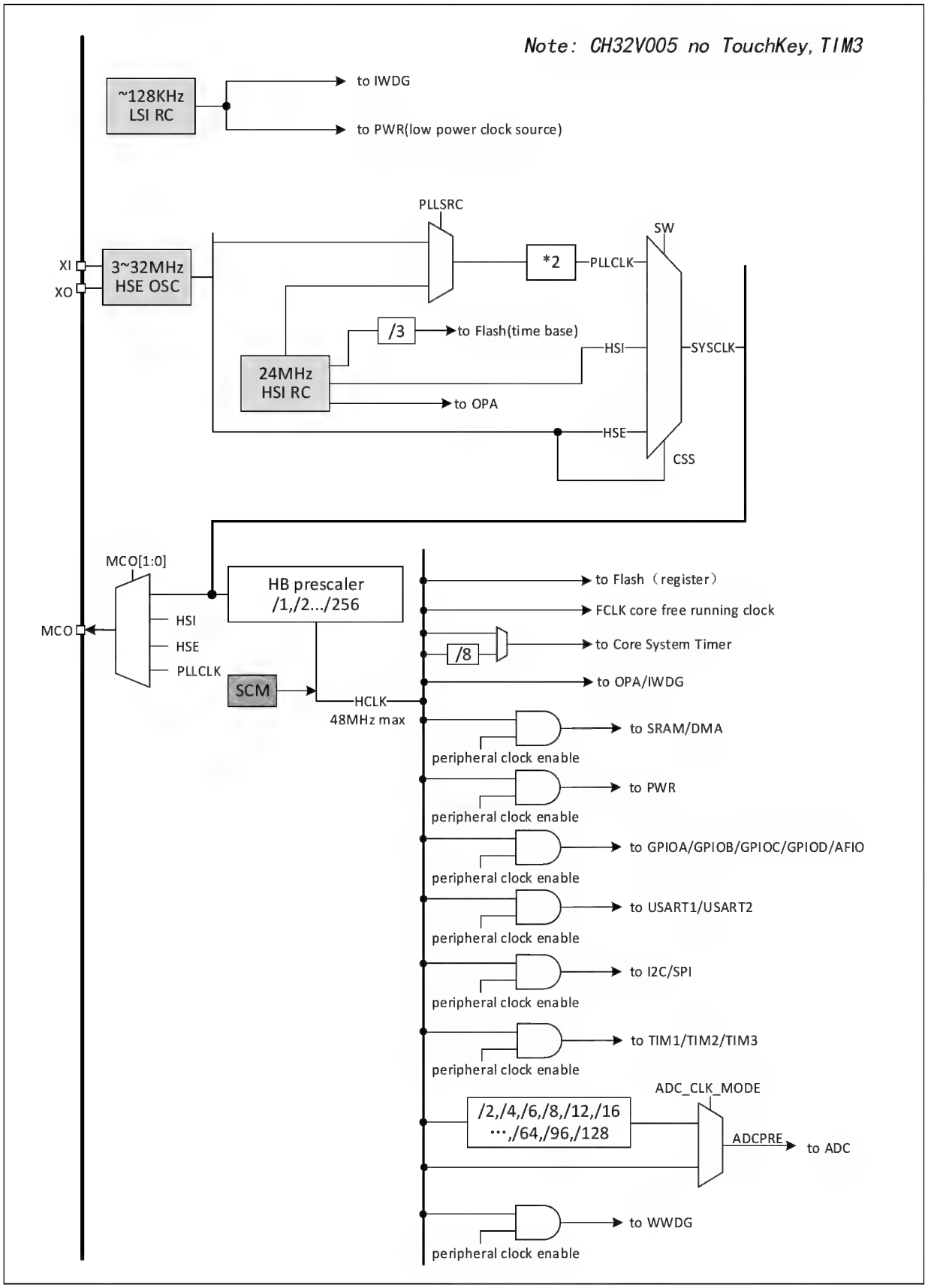
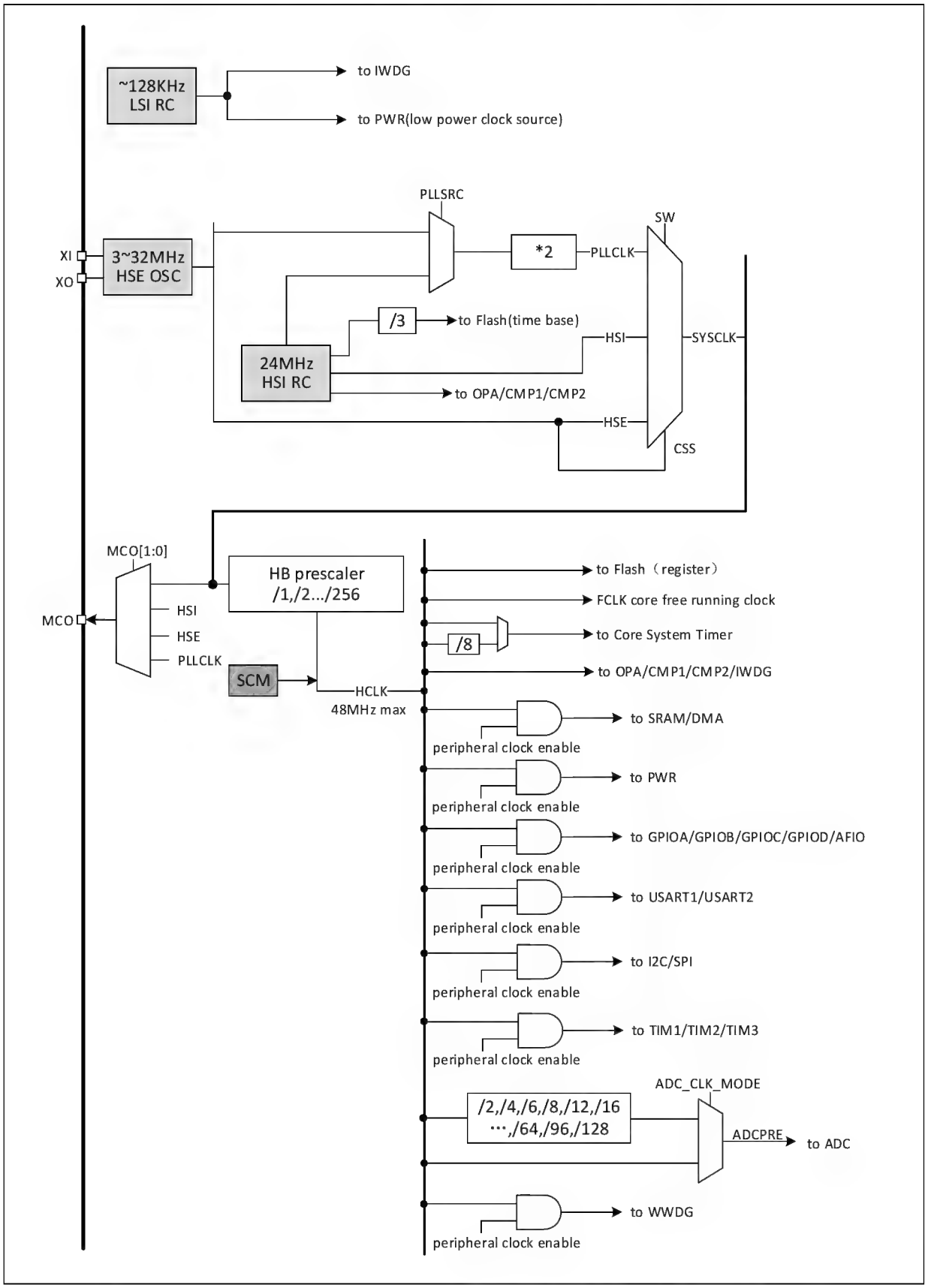


Figure 3-4 CH32V007, CH32M007 clock tree block diagram



3.3.2 High-speed Clock (HSI/HSE)

HSI is a high-speed clock signal generated by the system's internal 24MHz RC oscillator. HSI RC oscillator can provide system clock without any external devices. It has a short start-up time. HSI is enabled and disabled by setting the HSION bit in the RCC_CTLR register, and the HSIRDY bit indicates whether the HSI RC oscillator is stable or not. The system defaults HSION and HSIRDY to 1 (It is recommended not to turn them off). If the HSIRDYIE bit in the RCC_INTR register is set, the corresponding interrupt will be generated.

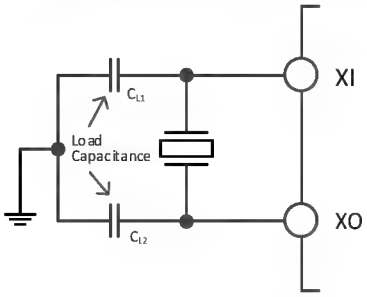
- Factory calibration: The difference of manufacturing process will cause different RC oscillation frequency for each chip, so HSI calibration is performed for each chip before it is shipped. After system reset, the factory calibration value is loaded into HSICAL[7:0] of the RCC_CTLR register.
- User tuning: Based on different voltages or ambient temperatures, the application can adjust the HSI frequency by using the HSITRIM[4:0] bits in the RCC_CTLR register.

Note: If the HSE crystal oscillator fails, the HSI clock is used as a backup clock source (Clock safety system).

HSE is an external high speed clock signal, including external crystal/ceramic resonator generation or external high speed clock feed.

- External crystal / ceramic resonator (HSE crystal): external oscillator provides a more accurate clock source for the system. For further information, please refer to the electrical characteristics section of the data manual. The HSE crystal can be turned on and off by setting the HSEON bit in the RCC_CTLR register, and the HSERDY bit indicates whether the HSE crystal oscillation is stable or not, and the hardware sends the clock into the system after HSERDY position 1. If the HSERDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

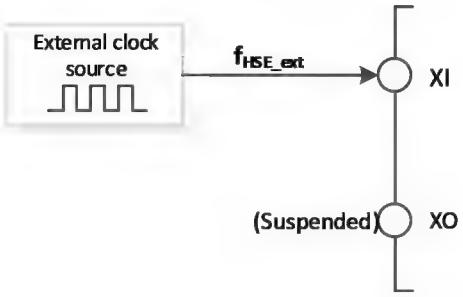
Figure 3-5 High-speed external crystal circuit



Note: The load capacitance should be as close to the oscillator pin as possible, and the capacity value should be selected according to the parameters of the crystal manufacturer.

- External high-speed clock source (HSE bypass): this mode feeds the clock source directly from the outside to the XI pin, and the XO pin is suspended. The application needs to set the HSEBYP bit when the HSEON bit is 0, turn on the HSE bypass function, and then set the HSEON bit.

Figure 3-6 High-speed clock source circuit



3.3.3 Low-speed Clock (LSI)

LSI is a low-speed clock signal generated by a RC oscillator of about 128kHz in the system. It can keep running in standby mode, providing a clock reference for independent watchdogs and wake-up units. For further information, please refer to the electrical characteristics section of the data manual. LSI can be turned on and off by setting the LSION bit in the RCC_RSTSCKR register, and then check whether the LSIRC oscillation is stable by querying the LSIRDY bit, and the hardware sends the clock in after LSIRDY position 1. If the LSIRDYIE bit of the RCC_INTR register is set, the corresponding interrupt will be generated.

3.3.4 PLL Clock (PLLCLK)

The internal PLL clock can be selected from 2 clock sources by configuring the PLLSRC bit in the RCC_CFGR0 register, these settings must be done before the PLL is turned on, once the PLL is powered up these parameters cannot be altered. Setting the PLLON bit in the RCC_CTLR register is activated and deactivated, the PLLRDY bit indicates whether the PLL clock is stable or not, and the hardware sends the clock into the system only after PLL position 1. If the PLLRDYIE bit in the RCC_INTR register is set, the corresponding interrupt is generated.

PLL clock source:

- HSI clock
- HSE Clock

3.3.5 Bus/Peripheral Clock

3.3.5.1 System Clock (SYSCLK)

By configuring the RCC_CFGR0 register SW [1:0] bit to configure the system clock source, SWS [1:0] indicates the current system clock source.

- HSI as system clock
- HSE as system clock
- PLL as system clock

After a controller reset, the default HSI clock is selected as the system clock source. Switching between clock sources must occur only when the target clock source is ready.

3.3.5.2 HB Bus Peripheral Clock (HCLK)

You can configure the clock of the HB bus by configuring the HPRE [3:0] bit of the RCC_CFGR0 register. The bus clock determines that the peripheral interface mounted below it accesses the clock reference. The application can reduce the power consumption of some peripherals by adjusting different values.

Through each bit in the RCC_PB1PRSTR and RCC_PB2PRSTR registers, different peripheral modules can be reset and restored to the initial state.

The communication clock interface of different peripheral modules can be turned on or off separately through each bit in the RCC_HBPCENR, RCC_PB1PCENR and RCC_PB2PCENR registers. When using a peripheral, you first need to turn on its clock enable bit before you can access its register.

3.3.5.3 Independent Watchdog Clock

If the independent watchdog has been started by the hardware configuration or software, the LSI oscillator will be forced to open and cannot be turned off. After the LSI oscillator is stabilized, the clock is supplied to the IWDG.

3.3.5.4 Microcontroller Clock Output (MCO)

The microcontroller allows clock signals to be output to MCO pins. The multiplexing push-pull output mode is configured in the corresponding GPIO port register. by configuring the RCC_CFGR0 register MCO [2:0] bit, the following 4 clock signals can be selected as MCO clock output:

- System clock (SYSCLK) output
- HSI clock output
- HSE clock output
- PLL clock output

3.3.6 Clock Security System (CSS)

The clock security system is a running protection mechanism of the controller, which can switch to the HSI clock in the case of HSE clock transmission failure, and generate interrupt notification, allowing application software to complete the rescue operation.

Activate the clock security system by setting CSSON position 1 of the RCC_CTLR register. At this point, the clock monitor will be enabled after the HSE oscillator startup (HSERDY=1) delay and turned off after the HSE clock is turned off. Once the HSE clock fails during the operation of the system, the HSE oscillator will be turned off, the clock failure event will be sent to the brake input of the advanced timer (TIM1), and the clock security interrupt will be generated, CSSF position 1, and the application will enter the NMI unshielded interrupt. By setting the CSSC bit, the CSSF bit flag can be cleared and the NMI interrupt suspension bit can be revoked.

If the current HSE is the system clock, or the current HSE is the PLL input clock and the PLL is the system clock, the clock security system will automatically switch the system clock to the HSI oscillator and turn off the HSE oscillator and PLL in the event of a HSE failure.

3.3.7 System Clock Monitoring System (SCM)

The system clock monitoring system is a system clock monitoring mechanism of the controller. After the SYSCM_EN bit is turned on, if the system clock failure event occurs, the brake signal will be generated to the advanced timer TIM1, and the system clock failure flag (SYSCLK_FAILIF) will be set; if the SYSCLK_FAILIE bit is enabled in advance, it will enter the system clock failure interrupt.

3.4 Register Description

Table 3-1 RCC-related registers list

Name	Access address	Description	Reset value
R32_RCC_CTLR	0x40021000	Clock Control Register	0x0050XX83
R32_RCC_CFGR0	0x40021004	Clock Configuration Register	0x00000020
R32_RCC_INTR	0x40021008	Clock Interrupt Register	0x00000000
R32_RCC_PB2PRSTR	0x4002100C	PB2 Peripheral Reset Register	0x00000000
R32_RCC_PB1PRSTR	0x40021010	PB1 Peripheral Reset Register	0x00000000
R32_RCC_HBPCENR	0x40021014	HB Peripheral Clock Enable Register	0x00000004
R32_RCC_PB2PCENR	0x40021018	PB2 Peripheral Clock Enable Register	0x00000000
R32_RCC_PB1PCENR	0x4002101C	PB1 Peripheral Clock Enable Register	0x00000000
R32_RCC_RSTSCKR	0x40021024	Control/Status register	0x08000000

3.4.1 Clock Control Register (RCC_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						PLLRDY	PLLON	HSE_SI		SYSCM_EN	HSE_LP	CSSON	HSEBYP	HSERDY	HSEON

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					HSI_LP	HSIRDY	HSION

Bit	Name	Access	Description	Reset value
[31:26]	Reserved	RO	Reserved	0
25	PLLRDY	RO	PLL clock-ready lock flag bit. 1: PLL clock lock. 0: PLL clock is not locked.	0
24	PLLON	RW	PLL clock enable control bit. 1: Enable the PLL clock. 0: Disable the PLL clock. *Note: After entering Standby low-power mode, this bit is cleared by hardware to 0.*	0
[23:22]	HSE_SI	RW	HSE current supply regulating bit. When HSE_LP=0: 00: 1.6mA; 01: 2mA; 10: 2.5mA; 11: 1.3mA. When HSE_LP=1: 00: 800μA; 01: 1mA; 10: 1.25mA; 11: 1.5mA.	0
21	SYSCM_EN	RW	The system clock monitoring module SCM enable 1: Enable; 0: Disable.	0
20	HSE_LP	RW	HSE low-power mode on 1: On; 0: Off.	1
19	CSSON	RW	The clock security system enables control bits. 1: Enable clock security system. When HSE is ready (HSERDY is set to 1), the hardware turns on the clock monitoring function for HSE and finds that HSE abnormal triggers the CSSF flag and NMI interrupt; when HSE is not ready, the hardware turns off the clock monitoring function for HSE. 0: Disable the clock security system.	0

18	HSEBYP	RW	External high-speed crystal bypass control bit. 1: Bypass external high-speed crystal / ceramic resonator (Using external clock source) 0: No bypass high-speed external crystal / ceramic resonator. *Note: This bit needs to be written when HSEON is 0.*	0
17	HSERDY	RO	External high-speed crystal oscillation stable ready flag bit (Set by hardware). 1: External high-speed crystal oscillation is stable; 0: External high-speed crystal oscillation is not stable. *Note: After the HSEON bit is cleared, it needs 6 HSE cycles to clear 0.*	0
16	HSEON	RW	External high-speed crystal oscillates to enable the control bit. 1: Enable HSE oscillator; 0: Disable HSE oscillator. *Note: After entering the standby low-power mode, this bit is cleared by the hardware.*	0
[15:8]	HSICAL[7:0]	RO	The internal high-speed clock calibration value is initialized automatically when the system starts.	xxh
[7:3]	HSITRIM[4:0]	RW	Internal high-speed clock adjustment. The user can enter an adjustment value that is superimposed on the HSICAL [7:0] value to adjust the frequency of the internal HSIRC oscillator according to changes in voltage and temperature. The default value is 16, and the HSI can be adjusted to 24MHz±1%; the change of HSICAL at each step is adjusted to about 60kHz. 10000b	10000b
2	HSI_LP	RW	HSI low-power mode is turned on. 1: On; 0: Off.	0
1	HSIRDY	RO	Internal high-speed clock (24MHz) stable ready flag bit (Set by hardware). 1: Internal high-speed clock (24MHz) is stable; 0: Internal high-speed clock (24MHz) is not stable. *Note: After the HSION bit is cleared, it needs 6 HSI cycles to clear 0.*	1
0	HSION	RW	The internal high-speed clock (24MHz) enables control bits. 1: Enable HSI oscillator; 0: Disable HSI oscillator. *Note: When the external oscillator HSE returned from standby mode or used as the system clock fails, this bit is set to 1 by the hardware to start the RC oscillator of the*	1

			internal 24MHz.	

3.4.2 Clock Configuration Register 0 (RCC_CFGR0)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC_CLK_MODE	Reserved		ADC_CLK_ADJ	Reserved	MCO[2:0]			Reserved							PLL SRC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCPRE[4:0]					Reserved			HPRE[3:0]				SWS[1:0]		SW[1:0]	

Bit	Name	Access	Description	Reset value
31	ADC_CLK_MODE	RW	ADC clock mode: 1: HB clock does not divide frequency; 0: HB clock divides frequency.	0
[30:29]	Reserved	RO	Reserved	0
28	ADC_CLK_ADJ	RW	ADC clock duty cycle adjustment: 1: 3/4 high-level; 0: 1/2 high-level.	0
27	Reserved	RO	Reserved	0
[26:24]	MCO[2:0]	RW	Microcontroller MCO pin clock output control: 0xx: No clock output; 100: SYSCLK output; 101: RC oscillator clock (HSI) output of internal 24MHz; 110: External oscillator clock (HSE) output; 111: PLL clock output.	0
[23:17]	Reserved	RO	Reserved	0
16	PLLSRC	RW	Input clock source of PLL (Write only when PLL is turned off): 1: HSE feeds into PLL without frequency division; 0: HSI feeds into PLL without frequency division.	0
[15:11]	ADCPRE[4:0]	RW	ADC clock resource prescaler control: 000xx: HBCLK divided by 2 as ADC clock; 010xx: HBCLK divided by 4 as ADC clock; 100xx: HBCLK divided by 6 as ADC clock; 110xx: HBCLK divided by 8 as ADC clock; 00100: HBCLK divided by 4 as ADC clock; 01100: HBCLK divided by 8 as ADC clock; 10100: HBCLK divided by 12 as ADC clock; 11100: HBCLK divided by 16 as ADC clock; 00101: HBCLK divided by 8 as ADC clock; 01101: HBCLK divided by 16 as ADC clock; 10101: HBCLK divided by 24 as ADC clock;	0

			11101: HBCLK divided by 32 as ADC clock; 00110: HBCLK divided by 16 as ADC clock; 01110: HBCLK divided by 32 as ADC clock; 10110: HBCLK divided by 48 as ADC clock; 11110: HBCLK divided by 64 as ADC clock; 00111: HBCLK divided by 32 as ADC clock; 01111: HBCLK divided by 64 as ADC clock; 10111: HBCLK divided by 96 as ADC clock; 11111: HBCLK divided by 128 as ADC clock.	
[10:8]	Reserved	RW	Reserved	0
[7:4]	HPRE[3:0]	RW	HB clock resource prescaler control: 0000: Prescaler off; 0001: SYSCLK divided by 2; 0010: SYSCLK divided by 3; 0011: SYSCLK divided by 4; 0100: SYSCLK divided by 5; 0101: SYSCLK divided by 6; 0110: SYSCLK divided by 7; 0111: SYSCLK divided by 8; 1000: SYSCLK divided by 2; 1001: SYSCLK divided by 4; 1010: SYSCLK divided by 8; 1011: SYSCLK divided by 16; 1100: SYSCLK divided by 32; 1101: SYSCLK divided by 64; 1110: SYSCLK divided by 128; 1111: SYSCLK divided by 256.	0010b
[3:2]	SWS[1:0]	RO	SYSCLK state (Hardware set). 00: HSI as system clock source. 01: HSE as system clock source. 10: PLL as system clock source. 11: Not available.	0
[1:0]	SW[1:0]	RW	Select SYSCLK resource: 00: HSI as system clock source. 01: HSE as system clock source. 10: PLL as system clock source. 11: Not available. *Note: In the clock enabled Security system (CSSON=1), when the external oscillator HSE returned from standby or used as the system clock fails, the hardware forces the HSI to be selected as the system clock.*	0

3.4.3 Clock Interrupt Register (RCC_INTR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserved		PLLRDYC	HSERDYC	HSIRDYC	Reserved	LSIRDYC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PLLRDYIE	HSERDYIE	HSIRDYIE	SYSCLK_FAILIE	LSIRDYIE	CSSF	Reserved		PLLRDYF	HSERDYF	HSIRDYF	Reserved	LSIRDYF

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	CSSC	WO	Clear the clock security system interrupt flag bit (CSSF). 1: Clear the CSSF interrupt flag; 0: No action.	0
[22:21]	Reserved	RO	Reserved	0
20	PLLRDYC	WO	Clear PLL ready interrupt flag bit. 1: Clear PLLRDYF interrupt flag; 0: No action.	0
19	HSERDYC	WO	Clear HSE oscillator ready interrupt flag bit. 1: Clear HSERDYF interrupt flag; 0: No action.	0
18	HSIRDYC	WO	Clear HSI oscillator ready interrupt flag bit. 1: Clear HSIRDYF interrupt flag; 0: No action.	0
17	Reserved	RO	Reserved	0
16	LSIRDYC	WO	Clear LSI oscillator ready interrupt flag bit. 1: Clear LSIRDYF interrupt flag; 0: No action.	0
[15:13]	Reserved	RO	Reserved	0
12	PLLRDYIE	RW	PLL ready interrupt enable bit. 1: Enable PLL ready interrupt; 0: Disable PLL ready interrupt.	0
11	HSERDYIE	RW	HSE ready interrupt enable bit. 1: Enable HSE ready interrupt; 0: Disable HSE ready interrupt.	0
10	HSIRDYIE	RW	HSI ready interrupt enable bit. 1: Enable HSI ready interrupt; 0: Disable HSI ready interrupt.	0
9	SYSCLK_FAILIE	RW	SYSCLK invalid interrupt enable bit. 1: Enable SYSCLK invalid interrupt; 0: Disable SYSCLK invalid interrupt.	0
8	LSIRDYIE	RW	LSI ready interrupt enable bit. 1: Enable LSI ready interrupt;	0

			0: Disable LSI ready interrupt.	
7	CSSF	RO	Clock security system interrupt flag bit. 1: HSE clock failure, resulting in clock security interrupt CSSI; 0: No clock security system interrupt. Hardware setting, software write CSSC bit 1 clear.	0
[6:5]	Reserved	RO	Reserved	0
4	PLLRDYF	RO	PLL clock ready lock the interrupt flag. 1: PLL clock lock interrupt generation; 0: No PLL clock lock interrupt. Hardware setting, software write PLLRDYC bit 1 clear.	0
3	HSERDYF	RO	HSE clock ready interrupt flag. 1: HSE clock ready interrupt generation; 0: No HSE clock ready interrupt. Hardware setting, software write HSERDYC bit 1 clear.	0
2	HSIRDYF	RO	HSI clock ready interrupt flag. 1: HSI clock ready interrupt generation; 0: No HSI clock ready interrupt. Hardware setting, software write HSIRDYC bit 1 clear.	0
1	Reserved	RO	Reserved	0
0	LSIRDYF	RO	LSI clock ready interrupt flag. 1: LSI clock ready interrupt generation; 0: No LSI clock ready interrupt. Hardware setting, software write LSIRDYC bit 1 clear.	0

3.4.4 PB2 Peripheral Reset Register (RCC_PB2PRSTR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1RST	USART2RST	SPI1RST	TIM1RST	Reserved	ADCRST	Reserved			IOPDRST	IOPCRST	IOPBRST	IOPARST	Reserved	AFIORST

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	USART1RST	RW	USART1 interface reset control. 1: Reset module; 0: No effect.	0
13	USART2RST	RW	USART2 interface reset control. 1: Reset module; 0: No effect.	0
12	SPI1RST	RW	SPI1 interface reset control. 1: Reset module; 0: No effect.	0
11	TIM1RST	RW	TIM1 module reset control.	0

			1: Reset module; 0: No effect.	
10	Reserved	RO	Reserved	0
9	ADCRST	RW	ADC module reset control. 1: Reset module; 0: No effect.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDRST	RW	PD port module reset control for I/O. 1: Reset module; 0: No effect.	0
4	IOPCRST	RW	PC port module reset control for I/O. 1: Reset module; 0: No effect.	0
3	IOPBRST	RW	PB port module reset control for I/O. 1: Reset module; 0: No effect.	0
2	IOPARST	RW	PA port module reset control for I/O. 1: Reset module; 0: No effect.	0
1	Reserved	RO	Reserved	0
0	AFIORST	RW	I/O auxiliary function module reset control. 1: Reset module; 0: No effect.	0

3.4.5 PB1 Peripheral Reset Register (RCC_PB1PRSTR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWR RST	Reserved						I2C1 RST	Reserved				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WW DG RST	Reserved								TIM3 RST	Reser ved	TIM2 RST

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	PWRRST	RW	Power interface module reset control. 1: Reset module; 0: No effect.	0
[27:22]	Reserved	RO	Reserved	0
21	I2C1RST	RW	I2C1 interface reset control. 1: Reset module; 0: No effect.	0
[20:12]	Reserved	RO	Reserved	0
11	WWDGRST	RW	Window watchdog reset control. 1: Reset module; 0: No effect.	0
[10:3]	Reserved	RO	Reserved	0
2	TIM3RST	RW	Timer 3 module reset control. 1: Reset module; 0: No effect.	0
1	Reserved	RO	Reserved	0
0	TIM2RST	RW	Timer 2 module reset control. 1: Reset module; 0: No effect.	0

3.4.6 HB Peripheral Clock Enable Register (RCC_HBPCENR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													SRAM EN	Reserved	DMA1 EN

Bit	Name	Access	Description	Reset value
[31:3]	Reserved	RO	Reserved	0
2	SRAMEN	RW	SRAM interface module clock enable bit. 1: SRAM interface module clock on in Sleep mode; 0: SRAM interface module clock off in Sleep mode.	1
1	Reserved	RO	Reserved	0
0	DMA1EN	RW	DMA1 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0

3.4.7 PB2 Peripheral Clock Enable Register (RCC_PB2PCENR)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	USART1 EN	USART2 EN	SPI1 EN	TIM1 EN	Reserved	ADC EN	Reserved			IOPD EN	IOPC EN	IOPB EN	IOPA EN	Reserved	AFIO EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	USART1EN	RW	USART1 interface module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
13	USART2EN	RW	USART2 interface module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
12	SPI1EN	RW	SPI1 interface module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
11	TIM1EN	RW	TIM1 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
10	Reserved	RO	Reserved	0
9	ADCEN	RW	ADC module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[8:6]	Reserved	RO	Reserved	0
5	IOPDEN	RW	PD port module clock enable bit for I/O.	0

			1: Module clock is on; 0: Module clock is off.	
4	IOPCEN	RW	PC port module clock enable bit for I/O. 1: Module clock is on; 0: Module clock is off.	0
3	IOPBEN	RW	PB port module clock enable bit for I/O. 1: Module clock is on; 0: Module clock is off.	0
2	IOPAEN	RW	PA port module clock enable bit for I/O. 1: Module clock is on; 0: Module clock is off.	0
1	Reserved	RO	Reserved	0
0	AFIOEN	RW	I/O auxiliary function module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0

3.4.8 PB1 Peripheral Clock Enable Register (RCC_PB1PCENR)

Offset address: 0x1C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved			PWREN	Reserved						I2C1EN	Reserved				

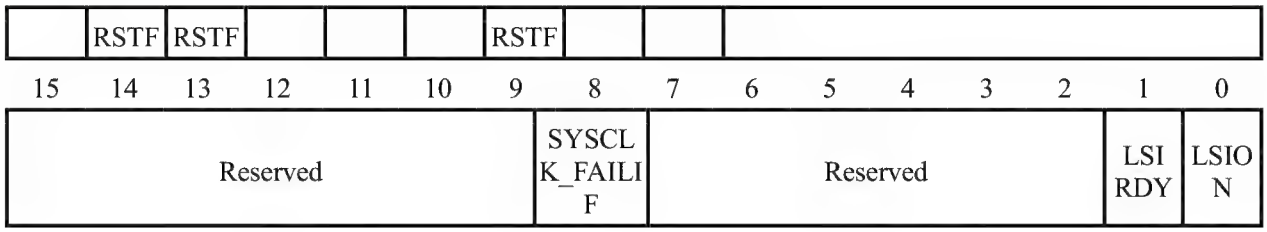
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				WWDGEN	Reserved								TIM3EN	Reserved	TIM2EN

Bit	Name	Access	Description	Reset value
[31:29]	Reserved	RO	Reserved	0
28	PWREN	RW	Power interface module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[27:22]	Reserved	RO	Reserved	0
21	I2C1EN	RW	I2C1 interface clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[20:12]	Reserved	RO	Reserved	0
11	WWDGEN	RW	Window watchdog clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
[10:3]	Reserved	RO	Reserved	0
2	TIM3EN	RW	Timer 3 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0
1	Reserved	RO	Reserved	0
0	TIM2EN	RW	Timer 2 module clock enable bit. 1: Module clock is on; 0: Module clock is off.	0

3.4.9 Control/Status Register (RCC_RSTSCKR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	WWDG	IWDG	SFTRSTF	PORRSTF	PINRSTF	OPCM	RMVF	ADCRSTF	Reserved						



Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved	
30	WWDGRSTF	RO	Window watchdog reset flag. 1: Window watchdog reset occurs; 0: No window watchdog reset occurs. When the window watchdog resets, the hardware sets 1; the software writes the RMVF bit to clear.	0
29	IWDGRSTF	RO	Independent watchdog reset flag. 1: Independent watchdog reset occurs; 0: No independent watchdog reset occurs. When the independent watchdog resets, the hardware sets 1; the software writes the RMVF bit to clear.	0
28	SFTRSTF	RO	Software reset flag. 1: Software reset occurs; 0: No software reset occurs. When the software resets, the hardware sets 1; the software writes the RMVF bit to clear.	0
27	PORRSTF	RO	Power-on/power-down reset flag. 1: Power-on/power-down reset occurs; 0: No power on/power off reset occurs. When power-on/power-down reset occurs, the hardware sets 1; the software writes the RMVF bit to clear.	1
26	PINRSTF	RO	External manual reset (NRST pin) flag. 1: NRST pin reset occurs; 0: No NRST pin reset occurs. When the NRST pin reset occurs, the hardware sets 1; the software writes the RMVF bit clear.	0
25	OPCMRSTF	RW	OPA_CMP reset flag. 1: OPA_CMP reset flag; 0: No effect.	0
24	RMVF	RW	Clear reset flag control. 1: Clear the reset flag; 0: No effect.	0
23	ADCRSTF	RW	ADC reset flag. 1: ADC reset flag; 0: No effect.	0
[22:9]	Reserved	RO	Reserved	0
8	SYSCLK_FAILI	RW0	System clock failure flag.	0

	F		1: System clock failure event occurs; 0: No system clock failure event occurs.	
[7:2]	Reserved	RO	Reserved	0
1	LSIRDY	RO	Internal low-speed clock (LSI) stable ready flag bit (set by hardware). 1: Internal low-speed clock (128kHz) is stable; 0: Internal low-speed clock (128kHz) is not stable. *Note: After the LSION bit is cleared, it needs 3 LSI cycles to clear 0.*	0
0	LSION	RW	Internal low-speed clock (LSI) enable control bit. 1: Enable LSI (128kHz) oscillator; 0: Disable LSI (128kHz) oscillator.	0

Note: Except that the reset flag can only be cleared by power-on reset, the rest can be removed by system reset.

Chapter 4 Independent Watchdog (IWDG)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The system is equipped with an independent watchdog (IWDG) to detect software failures caused by logic errors and external environment interference. The IWDG clock source comes from LSI and can be run independently of the main program, so it is suitable for situations with low precision requirements.

4.1 Main Features

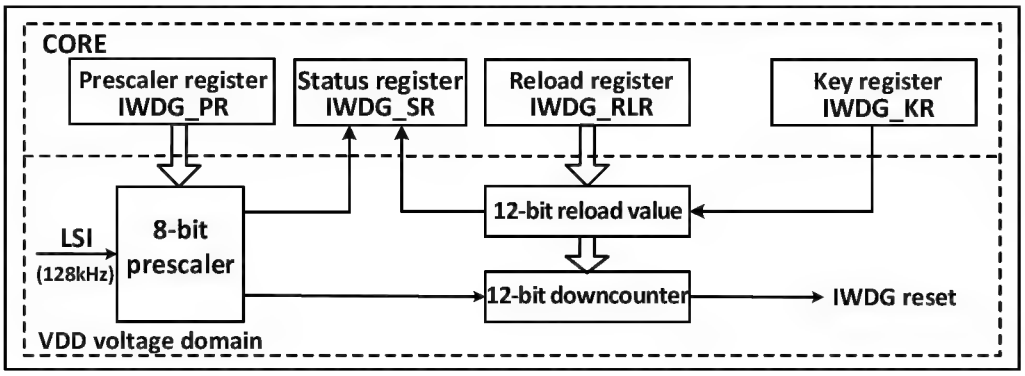
- 12-bit self-subtracting counter
- Clock source LSI divider, can run in low-power mode
- Reset condition: Counter value is reduced to 0

4.2 Function Description

4.2.1 Principle and Application

Independent watchdog clock source LSI clock, its function can still work in standby mode. When the watchdog counter decrements itself to 0, a system reset will be generated, so the timeout is (Reload value + 1) clock.

Figure 4-1 Block diagram of the structure of the independent watchdog



- Enable independent watchdog

After the system is reset, the watchdog is off, and the watchdog is turned on by writing 0xCCCC to the IWDG_CTLR register. After that, it can no longer be turned off unless a reset occurs.

If the hardware independent watchdog enable bit (IWDG_SW) is turned on in the user option byte, the IWDG will be permanently turned on after the microcontroller resets.

- Watchdog configuration

Inside the watchdog is a 12-bit counter running progressively. When the value of the counter is reduced to 0, a system reset will occur. To enable the IWDG function, you need to perform the following actions:

1) Counting time base: IWDG clock source LSI, through the IWDG_PSCR register to set the LSI frequency division value clock as the IWDG counting time base. Operation method: first write 0x5555 to the IWDG_CTLR register, and then modify the frequency division value in the IWDG_PSCR register. The PVU bit in the IWDG_STATR register indicates the update status of the frequency division value, and the frequency division value can only be modified and read out when the update is completed.
2) Reload value: used to update the current value of the counter in the independent watchdog, and the counter is

decremented by this value. Method of operation: first write 0x5555 to the IWDG_CTLR register, then modify the IWDG_RLDR register to set the target reload value. The RVU bit in the IWDG_STATR register indicates the update status of the reload value, and the IWDG_RLDR register can be modified and read out only after the update is completed.

3) Watchdog enable: write 0xCCCC to the IWDG_CTLR register to turn on the watchdog function.
4) Feed the dog: that is, before the watchdog counter decreases to 0, refresh the current counter value to prevent system reset. Write 0xAAAA to the IWDG_CTLR register and have the hardware update the IWDG_RLDR register value to the watchdog counter. This action needs to be performed regularly after the watchdog function is turned on, otherwise the watchdog reset action will occur.

4.2.2 Debug Mode

When the system enters debug mode, the IWDG counter can be configured by the debug module register to continue to work or stop.

4.3 Register Description

Table 4-1 IWDG-related registers list

Name	Access address	Description	Reset value
R16_IWDG_CTLR	0x40003000	Control register	0x0000
R16_IWDG_PSCR	0x40003004	Prescaler register	0x0000
R16_IWDG_RLDR	0x40003008	Reload value register	0x0FFF
R16_IWDG_STATR	0x4000300C	Status register	0x0000

4.3.1 Control Register (IWDG_CTLR)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	KEY[15:0]	WO	Operate the key value lock. 0xAAAA: Feed the dog. Load the IWDG_RLDR register value into the independent watchdog counter 0x5555: Allow modification of the R16_IWDG_PSCR and R16 IWDG _ RLDR registers 0xCCCC: Starts the watchdog, which is not restricted if the hardware watchdog is enabled (user option byte configuration).	0

4.3.2 Prescaler Register (IWDG_PSCR)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	PR[2:0]

Bit	Name	Access	Description	Reset value
[15:3]	Reserved	RO	Reserved	0
[2:0]	PR[2:0]	RW	IWDG clock prescaler factor, write 0x5555 to KEY before modifying this field. 000: Divided by 4; 001: Divided by 8; 010: Divided by 16; 011: Divided by 32; 100: Divided by 64; 101: Divided by 128; 110: Divided by 256; 111: Divided by 256. IWDG counting time base = LSI/divide factor. *Note: Before reading the value of this field, make sure the PVU bit in the IWDG_STATR register is 0, otherwise the read value is invalid.*	0

4.3.3 Reload Register (IWDG_RLDR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				RL[11:0]											

Bit	Name	Access	Description	Reset value
[15:12]	Reserved	RO	Reserved	0
[11:0]	RL[11:0]	RW	Counter reload value. Write 0x5555 to KEY before modifying this field. When 0xAAAA is written to KEY, the value of this field is loaded into the counter by the hardware, and the counter is then decremented from that value. *Note: Before reading and writing the field value, make sure that the RVU bit in the IWDG_STATR register is 0, otherwise it is invalid to read and write this field.*	FFFh

4.3.4 Status Register (IWDG_STATR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														RVU	PVU

Bit	Name	Access	Description	Reset value
[15:2]	Reserved	RO	Reserved	0
1	RVU	RO	Reload value updates the flag bit. Hardware setting or clearing 0.	0

			1: Reload value update in progress; 0: Reload update ends (up to 5 LSI cycles). *Note: The reload value register IWDG_RLDR can be read and written only after the RVU bit has been cleared.*	
0	PVU	RO	Clock frequency division coefficient updates flag bits. Hardware setting or clearing 0. 1: Clock division value update is in progress; 0: Clock division value update ends (Up to 5 LSI cycles). *Note: The frequency division factor register IWDG_PSCR can be read and written only after the PVU bit has been cleared.*	0

Note: After the pre-division or reinstallation value is updated, you do not have to wait for RVU or PVU to reset, you can continue to execute the following code. (This write operation continues to be completed even in low power mode)

Chapter 5 Window Watchdog (WWDG)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The window watchdog is generally used to monitor the software failures of the system, such as external interference, unforeseen logic errors and so on. It needs to refresh the counter (Feed the dog) within a specific window time (With upper and lower limits), otherwise the watchdog circuit will produce a system reset earlier or later than this window time.

5.1 Main Features

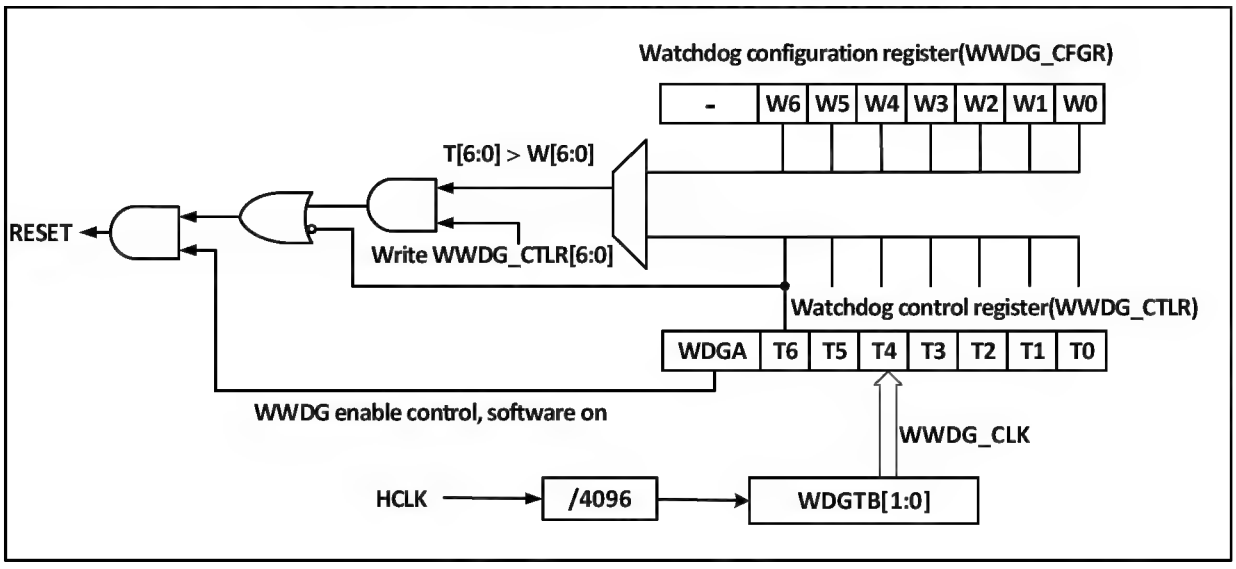
- Programmable 7-bit self-subtractive counter
- Double conditional reset: the current counter value is less than 0x40, or the counter value is reloaded outside the window time
- Wake up advance notice function (EWI), used to feed the dog in time to prevent system reset

5.2 Function Description

5.2.1 Principle and Application

The window watchdog runs based on a 7-bit decrement counter, which is mounted on the HB bus to count the frequency division of the time-based WWDG _ CLK source (HCLK/4096) clock, and the frequency division factor is set in the WDGTB [1:0] field in the configuration register WWDG_CFGR. The decrement counter is in a state of free operation, regardless of whether the watchdog function is turned on or not, the counter has been cyclically decreasing counting. As shown in figure 5-1, the internal structure block diagram of the window watchdog.

Figure 5-1 Block diagram of Window Watchdog structure



- Enable window watchdog

After the system is reset, the watchdog is off, the WDGA bit of the WWDG_CTLR register is set to turn on the watchdog, and then it can no longer be turned off unless a reset occurs.

Note: You can turn off the clock source of the WWDG by setting the RCC_PB1PCENR register, pause the WWDG_CLK count, stop the watchdog function indirectly, or reset the WWDG module by setting the RCC_PB1PRSTR register, which is equivalent to a reset.

- Watchdog configuration

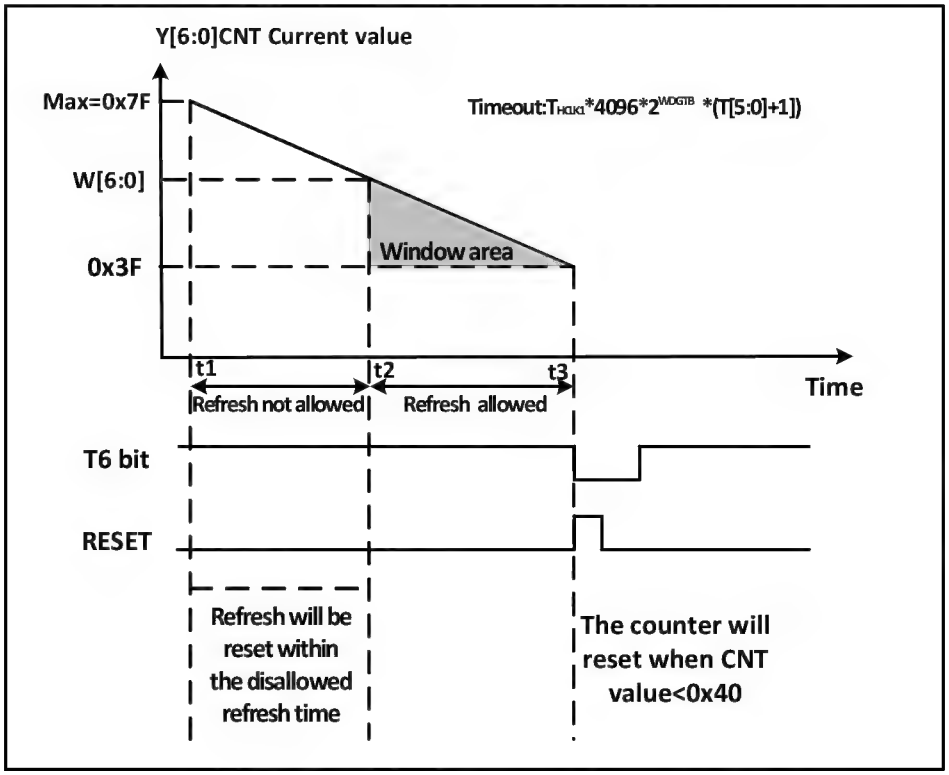
Inside the watchdog is a 7-bit counter running in a continuous cycle, which supports read and write access. To use the watchdog reset function, you need to perform the following actions:

1) Count time base: through the WDGTB [1:0] bit field of the WWDG_CFGR register, be careful to turn on the WWDG module clock of the RCC unit.
2) Window counter: sets the W [6:0] bit field of the WWDG_CFGR register. This counter is used by hardware to compare with the current counter. The value is configured by the user's software and will not be changed. As the upper limit of the window time.
3) Watchdog enable: WWDG_CTLR register WDGA bit software set 1, turn on watchdog function, you can reset the system.
4) Feed the dog: that is, refresh the current counter value and configure the T [6:0] bit field of the WWDG_CTLR register. This action needs to be performed within the periodic window time after the watchdog function is turned on, otherwise the watchdog reset action will occur.

- Feed the dog window time

As shown in figure 5-2, the grey area is the monitoring window area of the window watchdog, and its upper limit time t2 corresponds to the time point at which the counter value reaches the window value W[6:0], and its lower limit time t3 corresponds to the time point at which the counter value reaches 0x3F. In this area, t2 < t < t3 can carry out dog feeding operation (write T[6:0]) to refresh the value of the current counter.

Figure 5-2 Counting mode of Window Watchdog



- Watchdog reset

1) When the dog is not fed in time, resulting in the value of the T[6:0] counter from 0x40 to 0x3F, a "window watchdog reset" will appear, resulting in a system reset. That is, if the T6-bit is detected as 0 by the hardware, a system reset will occur.

Note: The application program can write T6-bit to 0 through software to realize system reset and equivalent software reset function.

2) When the counter refresh action is performed within the dog feeding time, that is, the write T [6:0] bit field is operated within the t1≤t≤t2 time, a "window watchdog reset" will appear, resulting in a system reset.

- Wakeup in advance

To prevent the system from resetting due to failure to refresh the counter in time, the watchdog module provides early wake-up interrupt (EWI) notification. When the counter is reduced to 0x40, an early wake-up signal is generated, and the EWIF flag is set to 1. If the EWI bit is set, the window watchdog will be triggered to interrupt at the same time. At this point, there is a counter clock cycle (Self-reduced to 0x3F) from the hardware reset, during which the application can immediately feed the dog.

5.2.2 Debug Mode

When the system enters debug mode, the WWDG counter can be configured by the debug module register to continue to work or stop.

5.3 Register Description

Table 5-1 WWDG-related registers list

Name	Access address	Description	Reset value
R16_WWDG_CTLR	0x40002C00	Control Register	0x007F
R16_WWDG_CFGR	0x40002C04	Configuration Register	0x007F
R16_WWDG_STATR	0x40002C08	Status Register	0x0000

5.3.1 Control Register (WWDG_CTLR)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								WDGA	T[6:0]						

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	WDGA	RW1	WWDG reset enable bit. 1: Enable watchdog function (Generate reset signal); 0: Disable watchdog function. Software write 1 is on, but only hardware is allowed to clear 0 after reset.	0
[6:0]	T[6:0]	RW	7-bit self-subtracting counter, minus 1 per $4096*2^{WDGTB}$ HCLK cycle. When the counter is reduced from 0x40 to 0x3F, that is, when T6 jumps to 0, a watchdog reset is generated.	7Fh

5.3.2 Configuration Register (WWDG_CFGR)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						EWI	WDGTB[1:0]		W[6:0]						

Bit	Name	Access	Description	Reset value
[15:10]	Reserved	RO	Reserved	0
9	EWI	RW1	Wake up the interrupt enable bit in advance. If this position is 1, an interrupt occurs when the value of the counter reaches 0x40. This bit can only be reset by hardware after 0.	0
[8:7]	WDGTB[1:0]	RW	Frequency division selection of window watchdog clock 00: Divided by 1, count time base = HCLK/4096; 01: Divided by 2, count time base = HCLK/4096/2; 10: Divided by 4, count time base = HCLK/4096/4; 11: Divided by 8, count time base = HCLK/4096/8.	0
[6:0]	W[6:0]	RW	Window watchdog 7-digit window value. Used to compare with the value of the counter. Dog feeding can only be done when the value of the counter is less than the window value and greater than 0x3F.	7Fh

5.3.3 Status Register (WWDG_STATR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															EWIF

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	WO	Reserved	0
0	EWIF	RW0	Wake up the interrupt flag bit in advance. When the counter reaches 0x40, this bit is set by hardware and must be cleared by software. The user setting is invalid. Even if the EWI is not set, it will be set as usual when the event occurs.	0

Chapter 6 Interrupt and Events (PFIC)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The CH32V00X series has a built-in programmable fast interrupt controller (PFIC- Programmable Fast Interrupt Controller) that supports up to 255interrupt vectors. The current system manages 25 peripheral interrupt channels and 4 kernel interrupt channels, and the rest are retained.

6.1 Main features

6.1.1 PFIC Controller

- 25 peripheral interrupts, each interrupt request has independent trigger and mask control bits, with dedicated status bits
- Programmable multi-level interrupt nesting, maximum nesting depth 2 levels, hardware stack depth 2 levels
- Fast interrupt entry and exit mechanism, hardware automatic stacking and recovery
- Vector Table Free (VTF) interrupt response mechanism, 2-channel programmable direct access to interrupt vector addresses

6.2 System Timer

- CH32V00X Series

The core comes with a 32-bit add counter (SysTick) that supports HCLK or HCLK/8 as a time base with high priority and can be used as a time reference after calibration.

6.3 Vector Table of Interrupts and Exceptions

Table 6-1 CH32V00X series vector table

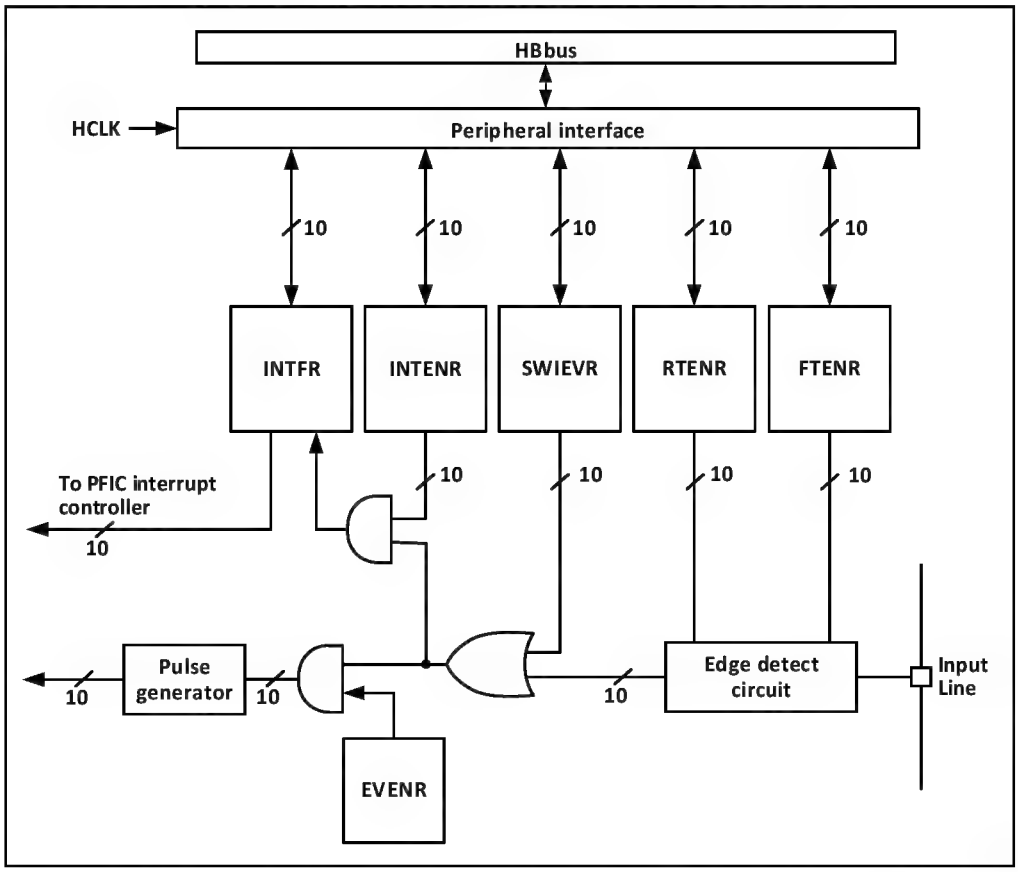
No.	Priority	Type	Name	Description	Entrance address
0	-	-	-	-	0x00000000
1	-	-	-	-	0x00000004
2	-2	Fixed	NMI	Non-maskable interrupts	0x00000008
3	-1	Fixed	HardFault	Abnormal interruptions	0x0000000C
4-11	-	-	-	Reserved	0x00000010-0x0000002C
12	0	Programmable	SysTick	System timer interrupt	0x00000030
13	-	-	-	Reserved	0x00000034
14	1	Programmable	SW	Software interrupt	0x00000038
15	-	-	-	Reserved	0x0000003C
16	2	Programmable	WWDG	Window timer interrupt	0x00000040
17	3	Programmable	PVD	Supply voltage detection interrupt (EXTI)	0x00000044
18	4	Programmable	FLASH	Flash global interrupt	0x00000048
19	5	Programmable	RCC	Reset and clock control interrupts	0x0000004C

20	6	Programmable	EXTI7_0	EXTI line 0-7 interrupt	0x00000050
21	7	Programmable	AWU	Wake-up interrupt	0x00000054
22	8	Programmable	DMA1_CH1	DMA1 channel 1 global interrupt	0x00000058
23	9	Programmable	DMA1_CH2	DMA1 channel 2 global interrupt	0x0000005C
24	10	Programmable	DMA1_CH3	DMA1 channel 3 global interrupt	0x00000060
25	11	Programmable	DMA1_CH4	DMA1 channel 4 global interrupt	0x00000064
26	12	Programmable	DMA1_CH5	DMA1 channel 5 global interrupt	0x00000068
27	13	Programmable	DMA1_CH6	DMA1 channel 6 global interrupt	0x0000006C
28	14	Programmable	DMA1_CH7	DMA1 channel 7 global interrupt	0x00000070
29	15	Programmable	ADC	ADC global Interrupt	0x00000074
30	16	Programmable	I2C1_EV	I2C1 event interrupt	0x00000078
31	17	Programmable	I2C1_ER	I2C1 error interrupt	0x0000007C
32	18	Programmable	USART1	USART1 global interrupt	0x00000080
33	19	Programmable	SPI1	SPI1 global Interrupt	0x00000084
34	20	Programmable	TIM1BRK	TIM1 brake interrupt	0x00000088
35	21	Programmable	TIM1UP	TIM1 update interrupt	0x0000008C
36	22	Programmable	TIM1TRG	TIM1 triggers an interrupt	0x00000090
37	23	Programmable	TIM1CC	TIM1 capture/compare interrupt	0x00000094
38	24	Programmable	TIM2	TIM2 global interrupt	0x00000098
39	25	Programmable	USART2	USART2 global interrupt	0x0000009C
40	26	Programmable	OPCM	OPCM global interrupt	0x000000A0

6.4 External Interrupt and Event Controller (EXTI)

6.4.1 Overview

Figure 6-1 External interrupt (EXTI) interface block diagram



As can be seen from figure 6-1, the trigger source of the external interrupt can be either the software interrupt (SWIEVR) or the actual external interrupt channel, and the signal of the external interrupt channel will first be screened by the edge detection circuit. As long as one of the software interrupts or external interrupt signals is generated, it will be output to both event enabling and interrupt enabling circuits through the OR gate circuit in the diagram. As long as an interrupt is enabled or an event is enabled, an interrupt or event will occur. The six registers of EXTI are accessed by the processor through the HB interface.

6.4.2 Wake-up Event

The system can wake up sleep patterns caused by WFE instructions through wake-up events. Wake-up events are generated through the following two configurations:

- Enable an interrupt in the register of the peripheral, but not in the PFIC of the core, and enable the SEVONPEND bit in the core. Reflected in EXTI, it enables EXTI interrupts, but does not enable EXTI interrupts in PFIC, while enabling SEVONPEND bits. When CPU wakes up from WFE, the interrupt flag bit and PFIC hang bit of EXTI need to be cleared.
- Enable an EXTI channel to be an event channel, and the CPU does not need to clear the interrupt flag bit and the PFIC hang bit after waking up from the WFE.

6.4.3 Description

The use of external interrupt needs to configure the corresponding external interrupt channel, that is, select the

appropriate trigger edge to enable the corresponding interrupt. When a set trigger edge appears on the external interrupt channel, an interrupt request will be generated and the corresponding interrupt flag bit will be set. Write 1 to the flag bit to clear it.

Use external hardware interrupt steps:

1) Configure GPIO operation
2) Configure the interrupt enable level (EXTI_INTENR) of the corresponding external interrupt channel
3) Configure trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger, or double edge trigger
4) Configure EXTI interrupts in the PFIC of the core to ensure that they respond correctly.

Use external hardware steps:

1) Configure GPIO operation
2) Configure the event enable level (EXTI_EVENR) of the corresponding external interrupt channel
3) Configure trigger edge (EXTI_RTENR or EXTI_FTENR), select rising edge trigger, falling edge trigger, or double edge trigger.

Use software interrupt/event steps:

1) Enable external interrupts (EXTI_INTENR) or external events (EXTI_EVENR)
2) If you use the interrupt service function, you need to set the EXTI interrupt in the PFIC of the core
3) Set software interrupt trigger (EXTI_SWIEVR), that is, an interrupt will occur.

6.4.4 External Event Mapping

Table 6-2 EXTI Interrupt Mapping

External interrupt/ Event lines	Mapping Event Description
EXTI0~EXTI7	Px0~Px7 (x=A/B/C/D), Any IO port can enable external interrupt/event function, which is configured by the AFIO_EXTICR register.
EXTI8	PVD event: voltage monitoring threshold exceeded.
EXTI9	Automatic wake-up event.

6.5 Register Description

6.5.1 EXTI Registers

Table 6-3 EXTI-related registers list

Name	Access address	Description	Reset value
R32_EXTI_INTENR	0x40010400	Interrupt enable register	0x00000000
R32_EXTI_EVENR	0x40010404	Event enable register	0x00000000
R32_EXTI_RTENR	0x40010408	Rising edge trigger enable register	0x00000000
R32_EXTI_FTENR	0x4001040C	Falling edge trigger enable register	0x00000000
R32_EXTI_SWIEVR	0x40010410	Soft interrupt event register	0x00000000
R32_EXTI_INTFR	0x40010414	Interrupt flag register	0x00000XXX

6.5.1.1 Interrupt Enable Register (EXTI_INTENR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	MRx	RW	The interrupt request signal of the external interrupt channel x is enabled. 1: Enable interrupts on this channel; 0: Mask interrupts on this channel.	0

6.5.1.2 Event Enable Register (EXTI_EVENR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	MRx	RW	The event request signal of the external interrupt channel x is enabled. 1: Enable interrupts on this channel; 0: Mask interrupts on this channel.	0

6.5.1.3 Rising Edge Trigger Enable Register (EXTI_RTENR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	TRx	RW	Enable the rising edge trigger of the external interrupt channel x. 1: Enable rising edge trigger of this channel;	0

			0: Disable rising edge trigger of this channel.	

6.5.1.4 Falling Edge Trigger Enable Register (EXTI_FTENR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	TRx	RW	Enable the falling edge trigger of the external interrupt channel x. 1: Enable falling edge trigger of this channel; 0: Disable falling edge trigger of this channel.	0

6.5.1.5 Software Interrupt Event Register (EXTI_SWIEVR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						SWIER 9	SWIER 8	SWIER 7	SWIER 6	SWIER 5	SWIER 4	SWIER 3	SWIER 2	SWIER 1	SWIER 0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	SWIERx	RW	A software interrupt is set on the corresponding external trigger interrupt channel. Setting here causes the interrupt flag bit (EXTI_INTFR) to correspond to the position bit, and if the interrupt enable (EXTI_INTENR) or event enable (EXTI_EVENR) is turned on, then an interrupt or event will occur.	0

6.5.1.6 Interrupt Flag Register (EXTI_INTFR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						IF9	IF8	IF7	IF6	IF5	IF4	IF3	IF2	IF1	IF0

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
[9:0]	IFx	RW1	The interrupt flag bit, which indicates that a corresponding external interrupt has occurred. Write 1 to clear this bit.	X

6.5.2 PFIC Registers

Table 6-4 List of PFIC-related registers

Name	Access address	Description	Reset value
R32_PFIC_ISR1	0xE000E000	PFIC Interrupt Enable Status Register 1	0x00000000
R32_PFIC_ISR2	0xE000E004	PFIC Interrupt Enable Status Register 2	0x00000000
R32_PFIC_IPR1	0xE000E020	PFIC Interrupt Pending Status Register 1	0x00000000
R32_PFIC_IPR2	0xE000E024	PFIC Interrupt Pending Status Register 2	0x00000000
R32_PFIC_ITHRESDR	0xE000E040	PFIC Interrupt Priority Threshold Configuration Register	0x00000000
R32_PFIC_CFGR	0xE000E048	PFIC Interrupt Configuration Register	0x00000000
R32_PFIC_GISR	0xE000E04C	PFIC Interrupt Global Status Register	0x00000000
R32_PFIC_VTFIDR	0xE000E050	PFIC VTF Interrupt ID Configuration Register	0x00000000
R32_PFIC_VTFADDRR0	0xE000E060	PFIC VTF Interrupt 0 Address Register	0x00000000
R32_PFIC_VTFADDRR1	0xE000E064	PFIC VTF Interrupt 1 Address Register	0x00000000
R32_PFIC_IENR1	0xE000E100	PFIC Interrupt Enable Setting Register 1	0x00000000
R32_PFIC_IENR2	0xE000E104	PFIC Interrupt Enable Setting Register 2	0x00000000
R32_PFIC_IRER1	0xE000E180	PFIC Interrupt Enable Clear Register 1	0x00000000
R32_PFIC_IRER2	0xE000E184	PFIC Interrupt Enable Clear Register 2	0x00000000
R32_PFIC_IPSR1	0xE000E200	PFIC Interrupt Pending Setting Register 1	0x00000000
R32_PFIC_IPSR2	0xE000E204	PFIC Interrupt Pending Setting Register 2	0x00000000
R32_PFIC_IPRR1	0xE000E280	PFIC Interrupt Pending Clear Register 1	0x00000000
R32_PFIC_IPRR2	0xE000E284	PFIC Interrupt Pending Clear Register 2	0x00000000
R32_PFIC_IACTR1	0xE000E300	PFIC Interrupt Activation Status Register 1	0x00000000
R32_PFIC_IACTR2	0xE000E304	PFIC Interrupt Activation Status Register 2	0x00000000
R32_PFIC_IPRIORx	0xE000E400	PFIC Interrupt Priority Configuration register	0x00000000
R32_PFIC_SCTLR	0xE000ED10	PFIC System Control Register	0x00000000

Note: 1. The default value for the PFIC_ISR1 register is 0xC, that is, NMI and exceptions are always enabled by default.

2. NMI and EXC support interrupting pending cleanup and setting operations, but not interrupting enable cleanup and setting operations.

6.5.2.1 PFIC Interrupt Enable Status Register 1 (PFIC_ISR1)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTENSTA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Reserved	INTENSTA14	Reserved	INTENSTA12	Reserved	INTENSTA3	INTENSTA2	Reserved

Bit	Name	Access	Description	Reset value
[31:16]	INTENSTA	RO	16#-31# Interrupt current enable status. 1: Current numbered interrupt is enabled. 0: Current numbered interrupt is not enabled.	0
15	Reserved	RO	Reserved	0
14	INTENSTA	RO	14# interrupt current enable status. 1: Current numbered interrupt is enabled. 0: Current numbered interrupt is not enabled.	0
13	Reserved	RO	Reserved	0
12	INTENSTA	RO	12# interrupt current enable status. 1: Current numbered interrupt is enabled. 0: Current numbered interrupt is not enabled.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	INTENSTA	RO	2#-3# interrupt current enable status. 1: Current numbered interrupt is enabled. 0: Current numbered interrupt is not enabled.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.2 PFIC Interrupt Enable Status Register 2 (PFIC_ISR2)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INTENSTA[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	INTENSTA	RO	32#-40# interrupt current enable status. 1: Current numbered interrupt is enabled. 0: Current numbered interrupt is not enabled.	0

6.5.2.3 PFIC Interrupt Pending Status Register 1 (PFIC_IPR1)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSTA[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PENDST	Reserved	PENDST	Reserved								PENDST	PENDST	Reserved	

	A14		A12		A3	A2	

Bit	Name	Access	Description	Reset value
[31:16]	PENDSTA	RO	16#-31# interrupt current pending status. 1: Current numbered interrupt is pending. 0: Current numbered interrupt is not pending.	0
15	Reserved	RO	Reserved	0
14	PENDSTA	RO	14# interrupt current pending status. 1: Current numbered interrupt is pending. 0: Current numbered interrupt is not pending.	0
13	Reserved	RO	Reserved	0
12	PENDSTA	RO	12# interrupt current pending status. 1: Current numbered interrupt is pending. 0: Current numbered interrupt is not pending.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSTA	RO	2#-3# interrupt current pending status. 1: Current numbered interrupt is pending. 0: Current numbered interrupt is not pending.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.4 PFIC Interrupt Pending Status Register 2 (PFIC_IPR2)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PENDSTA[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	PENDSTA	RO	32#-40# interrupt current pending status. 1: Current numbered interrupt is pending. 0: Current numbered interrupt is not pending.	0

6.5.2.5 PFIC Interrupt Priority Threshold Configuration Register (PFIC_ITHRESDR)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																								THRESHOLD[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	THRESHOLD[7:0]	RW	Interrupt priority threshold setting value. The interrupt priority value lower than the	0

			current setting value, when hung, does not perform interrupt service; this register is 0 means the threshold register function is invalid. [7:6]: priority threshold. [5:0]: reserved, fixed to 0, write invalid.	

6.5.2.6 PFIC Interrupt Configuration Register (PFIC_CFGR)

Offset address: 0x48

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYCODE[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RSTSYS	Reserved						

Bit	Name	Access	Description	Reset value
[31:16]	KEYCODE[15:0]	WO	Corresponding to different target control bits, the corresponding security access identification data needs to be written synchronously before it can be modified, and the readout data is fixed to 0. KEY1 = 0xFA05; KEY2 = 0xBCAF; KEY3 = 0xBEEF.	0
[15:8]	Reserved	RO	Reserved	0
7	RSTSYS	WO	System reset (Write KEY3 synchronously). Automatically clear 0. Write 1 is valid, write 0 is invalid. *Note: The same function as the SYSRST bit of the PFIC_SCTLR register.*	0
[6:0]	Reserved	RO	Reserved	0

6.5.2.7 PFIC Interrupt Global Status Register (PFIC_GISR)

Offset address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							GPENDSTA	GACTSTA	NESTSTA[7:0]						

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0

9	GPENDSTA	RO	Whether there are any interrupts currently pending. 1: Yes; 0: No.	0
8	GACTSTA	RO	Whether any interrupts are currently being executed. 1: Yes; 0: No.	0
[7:0]	NESTSTA[7:0]	RO	The current interrupt nesting state supports a maximum of level 2 nesting, and the maximum hardware stack depth is level 2. 0x03: Level 2 interrupt; 0x01: Level 1 interrupt; Other: No interrupt occurred.	0

6.5.2.8 PFIC VTF Interrupt ID Configuration Register (PFIC_VTFIDR)

Offset address: 0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VTFID1								VTFID0							

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:8]	VTFID1	RW	Configure the interrupt number for VTF interrupt 1.	0
[7:0]	VTFID0	RW	Configure the interrupt number for VTF interrupt 0.	0

6.5.2.9 PFIC VTF Interrupt 0 Address Register (PFIC_VTFADDRR0)

Offset address: 0x60

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR0[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR0[15:1]															VTF0EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR0[31:1]	RW	VTF interrupt 0 service program address bit [31:1], bit0 is 0.	0
0	VTF0EN	RW	VTF interrupts the 0 enable bit. 1: Enable VTF interrupt 0 channel; 0: Disable.	0

6.5.2.10 PFIC VTF Interrupt 1 Address Register (PFIC_VTFADDRR1)

Offset address: 0x64

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDR1[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR1[15:1]															VTF1EN

Bit	Name	Access	Description	Reset value
[31:1]	ADDR1[31:1]	RW	VTF interrupt 1 serve program address bit[31:1], bit0 is 0。	0
0	VTF1EN	RW	VTF interrupt 1 enable bit. 1: Enable VTF interrupt 1 channel; 0: Disable.	0

6.5.2.11 PFIC Interrupt Enable Setting Register 1 (PFIC_IENR1)

Offset address: 0x100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTEN[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INTEN14	Reserved	INTEN12	Reserved											

Bit	Name	Access	Description	Reset value
[31:16]	INTEN	WO	16#-31# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	INTEN	WO	14# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	INTEN	WO	12# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

6.5.2.12 PFIC Interrupt Enable Setting Register 2 (PFIC_IENR2)

Offset address: 0x104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INTEN[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	INTEN	WO	32#-40# interrupt enable control. 1: Current numbered interrupt enable; 0: No effect.	0

6.5.2.13 PFIC Interrupt Enable Clear Register 1 (PFIC_IRER1)

Offset address: 0x180

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INTRSET[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	INTRSET14	Reserved	INTRSET12	Reserved											

Bit	Name	Access	Description	Reset value
[31:16]	INTRSET	WO	16#-31# interrupt disable control. 1: Current numbered interrupt disable; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	INTRSET	WO	14# interrupt disable control. 1: Current numbered interrupt disable; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	INTRSET	WO	12# interrupt disable control. 1: Current numbered interrupt disable; 0: No effect.	0
[11:0]	Reserved	RO	Reserved	0

6.5.2.14 PFIC Interrupt Enable Clear Register 2 (PFIC_IRER2)

Offset address: 0x184

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									INTRSET[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	INTRSET	WO	32#-40# interrupt disable control.	0

			1: Current numbered interrupt disable; 0: No effect.	

6.5.2.15 PFIC Interrupt Pending Setup Register 1 (PFIC_IPSR1)

Offset address: 0x200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDSET[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PENDSET14	Reserved	PENDSET12	Reserved								PEND SET3	PEND SET2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	PENDSET	WO	16#-31# interrupt pending setting. 1: Current numbered interrupt pending; 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PENDSET	WO	14# interrupt pending setting. 1: Current numbered interrupt pending; 0: No effect.	0
13	Reserved	RO	Reserved	0
12	PENDSET	WO	12# interrupt pending setting. 1: Current numbered interrupt pending; 0: No effect.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDSET	WO	2#-3# interrupt pending setting. 1: Current numbered interrupt pending; 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.16 PFIC Interrupt Pending Setup Register 2 (PFIC_IPSR2)

Offset address: 0x204

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PENDSET[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	PENDSET	WO	32#-40# interrupt pending setting. 1: Current numbered interrupt pending; 0: No effect.	0

6.5.2.17 PFIC Interrupt Pending Clear Register 1 (PFIC_IPRR1)

Offset address: 0x280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PENDRST[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PENDRST14	Reserved	PENDRST12	Reserved								PENDRST3	PENDRST2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	PENDRST	WO	16#-31# interrupt pending clear. 1: The current numbered interrupt clears pending state. 0: No effect.	0
15	Reserved	RO	Reserved	0
14	PENDRST	WO	14# interrupt pending clear. 1: The current numbered interrupt clears pending state. 0: No effect.	0
13	Reserved	RO	Reserved	0
12	PENDRST	WO	12# interrupt pending clear. 1: The current numbered interrupt clears pending state. 0: No effect.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	PENDRST	WO	2#-3# interrupt pending clear. 1: The current numbered interrupt clears pending state. 0: No effect.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.18 PFIC Interrupt Pending Clear Register 2 (PFIC_IPRR2)

Offset address: 0x284

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									PENDRST[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	PENDRST	WO	32#-40# interrupt pending clear. 1: The current numbered interrupt clears	0

			pending state. 0: No effect.	

6.5.2.19 PFIC Interrupt Activation Status Register 1 (PFIC_IACTR1)

Offset address: 0x300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IACTS [31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IACTS14	Reserved	IACTS12	Reserved								IACTS3	IACTS2	Reserved	

Bit	Name	Access	Description	Reset value
[31:16]	IACTS	RO	16#-31# Interrupt execution status. 1: The current numbered interrupt in execution. 0: The current numbered interrupt is not executed.	0
15	Reserved	RO	Reserved	0
14	IACTS	RO	14# Interrupt execution status. 1: The current numbered interrupt in execution. 0: The current numbered interrupt is not executed.	0
13	Reserved	RO	Reserved	0
12	IACTS	RO	12# Interrupt execution status. 1: The current numbered interrupt in execution. 0: The current numbered interrupt is not executed.	0
[11:4]	Reserved	RO	Reserved	0
[3:2]	IACTS	RO	2#-3# Interrupt execution status. 1: The current numbered interrupt in execution. 0: The current numbered interrupt is not executed.	0
[1:0]	Reserved	RO	Reserved	0

6.5.2.20 PFIC Interrupt Activation Status Register 2 (PFIC_IACTR2)

Offset address: 0x304

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									IACTS[40:32]						

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0

[8:0]	IACTS	RO	32#-40# Interrupt execution status. 1: The current numbered interrupt in execution. 0: The current numbered interrupt is not executed.	0

6.5.2.21 PFIC Interrupt Priority Configuration Register (PFIC_IPRIORx) (x=0-63)

Offset address: 0x400-0x4FF

The controller supports 256 interrupts (0-255), each using 8 bits to set the control priority.

	31 24	23 16	15 8	7 0
IPRIOR63	PRIO_255	PRIO_254	PRIO_253	PRIO_252
…	…	…	…	…
IPRIORx	PRIO_（4x+3）	PRIO_（4x+2）	PRIO_（4x+1）	PRIO_（4x）
…	…	…	…	…
IPRIOR0	PRIO_3	PRIO_2	PRIO_1	PRIO_0

Bit	Name	Access	Description	Reset value
[2047:2040]	IP_255	RW	Same as IP_0 description.	0
…	…	…	…	…
[31:24]	IP_3	RW	Same as IP_0 description.	0
[23:16]	IP_2	RW	Same as IP_0 description.	0
[15:8]	IP_1	RW	Same as IP_0 description.	0
[7:0]	IP_0	RW	Number 0 interrupt priority configuration. [7:6]: priority control bits. If no nesting is configured, no preemption bits. Bit7 is preempted if 2 levels of nesting are configured. [5:0]: reserved, fixed to 0, write invalid.	0

6.5.2.22 PFIC System Control Register (PFIC_SCTLR)

Offset address: 0xD10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYSRST	Reserved														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved										SETEVENT	SEVONPEND	WFITOWFE	SLEEPDEEP	SLEEPONEXIT	Reserved

Bit	Name	Access	Description	Reset value

31	SYSRST	WO	System reset, 0 is cleared automatically. Write 1 is valid, write 0 is invalid, and it has the same effect as PFIC_CFGR register.	0
[30:6]	Reserved	RO	Reserved	0
5	SETEVENT	WO	Set the event, wake up the WFE situation.	0
4	SEVONPEND	RW	When an event or interrupt hang state occurs, the system can be woken up from after the WFE instruction, or if the WFE instruction has not been executed, the system will be woken up immediately after the next execution of the instruction. 1: Enabled events and all interrupts (including unopened interrupts) can wake up the system 0: Only enabled events and enabled interrupts can wake up the system.	0
3	WFITOWFE	RW	Execute the WFI instruction as if it were WFE. 1: Use the subsequent WFI instruction as the WFE instruction 0: No effect.	0
2	SLEEPDEEP	RW	Control the low power mode of the system. 1: deepsleep; 0: sleep。	0
1	SLEEPONEXIT	RW	Control the state of the system after leaving the interrupt service program. 1: The system enters low-power mode 0: The system enters the main program.	0
0	Reserved	RO	Reserved	0

6.5.3 Dedicated CSR Registers

Some Control and Status Registers (CSR) are defined in the RISC-V architecture to configure or identify or record the running status. CSR registers are internal registers within the kernel and use a dedicated 12-bit address space. In addition to the standard registers defined in the RISC-V privileged architecture document, the CH32V00X chip also adds some custom registers that need to be accessed by csr instructions.

Note: Such registers marked with the "MRW, MRO, MRW1" attribute need to be accessed by the system in machine mode.

6.5.3.1 Interrupt System Control Register (INTSYSCR)

CSR address: 0x804

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														INES	HWS

	T EN	TKE N

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	MRO	Reserved	0
1	INESTEN	MRW	Interrupt nesting enable. 1: Interrupt nesting function is enabled. 0: Interrupt nesting function is disabled.	0
0	HWSTKEN	MRW	Hardware stack enable. 1: Hardware stacking function is enabled. 0: Hardware stacking function is disabled.	0

6.5.3.2 Exception Entry Base Address Register (MTVEC)

CSR address: 0x305

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[15:2]														MODE 1	MOD E0

Bit	Name	Access	Description	Reset value
[31:2]	BASEADDR	MRW	Interrupt vector table base address.	0
1	MODE1	MRW	Interrupt vector scale recognition pattern. 1: Identified by absolute address, full range is supported, but must be redirected 0: Identified by jump instruction, limited range, support non-jump instruction.	0
0	MODE0	MRW	Interrupt or exception entry address mode selection. 1: Address offset according to interrupt number * 4 0: Use a unified entry address.	0

6.5.4 STK Register Description

Table 6-5 STK-related registers list

Name	Access address	Description	Reset value
R32_STK_CTLR	0xE000F000	System Count Control Register	0x00000000
R32_STK_SR	0xE000F004	System Count Status Register	0x00000000
R32_STK_CNTL	0xE000F008	System Counter Register	0x00000000
R32_STK_CMPLR	0xE000F010	Count Compare Register	0x00000000

6.5.4.1 System Count Control Register (STK_CTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWIE	Reserved														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved												STRE	STCLK	STIE	STE

Bit	Name	Access	Description	Reset value
31	SWIE	RW	Software interrupt trigger enable (SWI). 1: Triggering software interrupts. 0: Turn off the trigger. After entering software interrupt, software clear 0 is required, otherwise it is continuously triggered.	0
[30:4]	Reserved	RO	Reserved	0
3	STRE	RW	Auto-reload count enable bit. 1: Re-counting from 0 after counting up to the comparison value. 0: Count up to the comparison value and continue counting up, count down to 0 and start counting down again from the maximum value.	0
2	STCLK	RW	Counter clock source selection bit. 1: HCLK for time base. 0: HCLK/8 for time base.	0
1	STIE	RW	Counter interrupt enable control bit. 1: Enable counter interrupt. 0: Disable counter interrupt.	0
0	STE	RW	System counter enable control bit. 1: Turn on the system counter STK. 0: Turn off the system counter STK and the counter stops counting.	0

6.5.4.2 System Count Status Register (STK_SR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															CNTIF

Bit	Name	Access	Description	Reset value
[31:1]	Reserved	RO	Reserved	0
0	CNTIF	RW0	Count value comparison flag, write 0 to clear, write 1 to invalidate.	0

			1: Up count reaches the comparison value. 0: The comparison value is not reached.	

6.5.4.3 System Counter Register (STK_CNTL)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CNT	RW	The current counter count value is 32 bits.	0

6.5.4.4 Counting Comparison Register (STK_CMPLR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CMP	RW	Set the comparison counter value to 32 bits.	0

Chapter 7 GPIO and Alternate Function (GPIO/AFIO)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The GPIO port can be configured in a variety of input or output modes, with built-in pull-up or pull-down resistors that can be closed, and can be configured for push-pull or open-drain functions. The GPIO port can also be reused into other functions.

7.1 Main Features

Each pin of the port can be configured to one of the following multiple modes

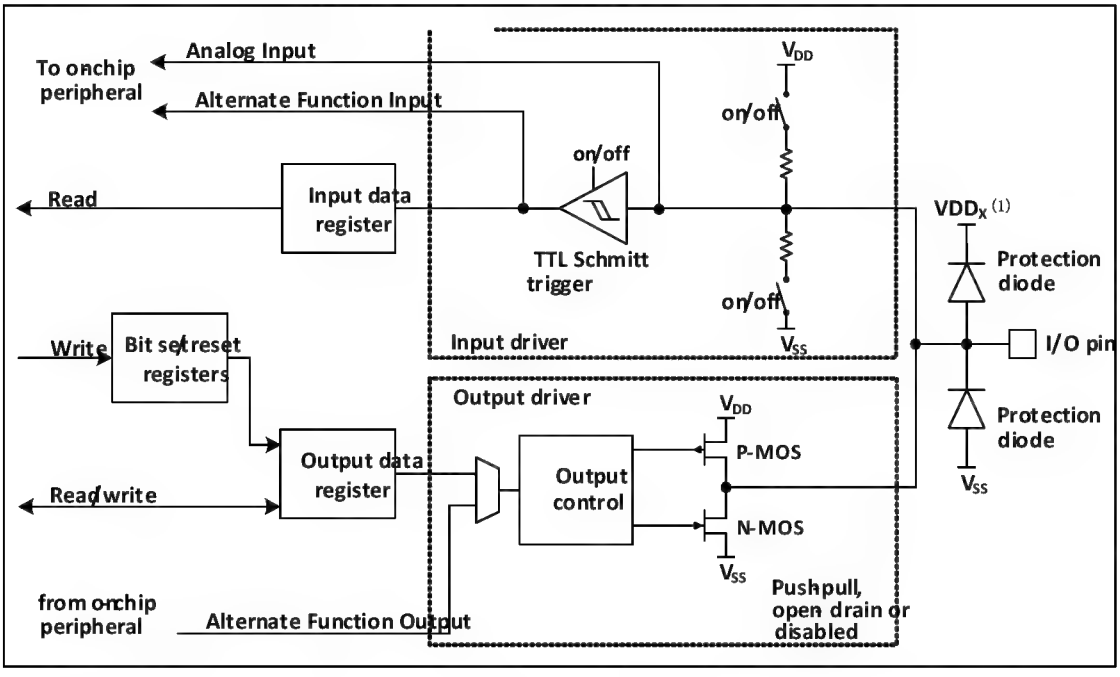
- Floating input
- Pull-up input
- Pull-down input
- Analog input
- Open-drain output
- Push-pull output
- Inputs and outputs of alternate functions

Many pins have the alternate function, and many other peripherals map their output and input channels to these pins. The specific usage of these alternate pins needs to refer to each peripheral. Whether these pins are reused and remapped is explained in this chapter.

7.2 Function Description

7.2.1 Overview

Figure 7-1 GPIO module basic structure block diagram



Note: (1) VDDx is VDD when GPIO is normal IO, and VDDx is VDD_FT when GPIO is FTIO.

As shown in figure 7-1, the IO port structure, each pin has two protection diodes inside the chip, and the IO port can be divided into input and output drive modules. The input driver has a weak pull-up resistor, which can be connected to AD and other analog input peripherals; if you input to a digital peripheral, you need to go through a

TTL Schmitt trigger, and then connect to the GPIO input register or other multiplexed peripherals. The output driver has a pair of MOS tubes, and the IO port can be configured to open-drain or push-pull output by configuring whether the upper and lower MOS tubes are enabled or not; the output driver can also be configured to be controlled by GPIO or by other peripherals that are reused.

7.2.2 GPIO Initialization Function

Just after the reset, the GPIO port is running in the initial state, at this time, most IO ports are running in the floating input state, but there are also HSE and other peripheral-related pins that run on the peripheral reuse function. For specific initialization functions, please refer to the relevant sections of the pin description.

7.2.3 External Interrupts

All GPIO ports can be configured with external interrupt input channels, but an external interrupt input channel can only be mapped to one GPIO pin, and the sequence number of the external interrupt channel must be consistent with the tag of the GPIO port, such as PA1 (Or PC1, PD1, etc.) can only be mapped to EXTI1, and EXTI1 can only accept the mapping of one of PA1, PC1 or PD1, etc., both sides are one-to-one relationship.

7.2.4 Alternate Functions

When using the alternate function, you must pay attention to:

- Using the alternate function of the input direction, the port must be configured in the alternate input mode, and the up and down settings can be set according to the actual needs.
- Using the alternate function in the output direction, the port must be configured in the multiplexing output mode, and push-pull or open-drain can be set according to the actual situation.
- For bi-directional alternate, the port must be configured for multiplexed output mode, in which case the driver is configured for floating input mode

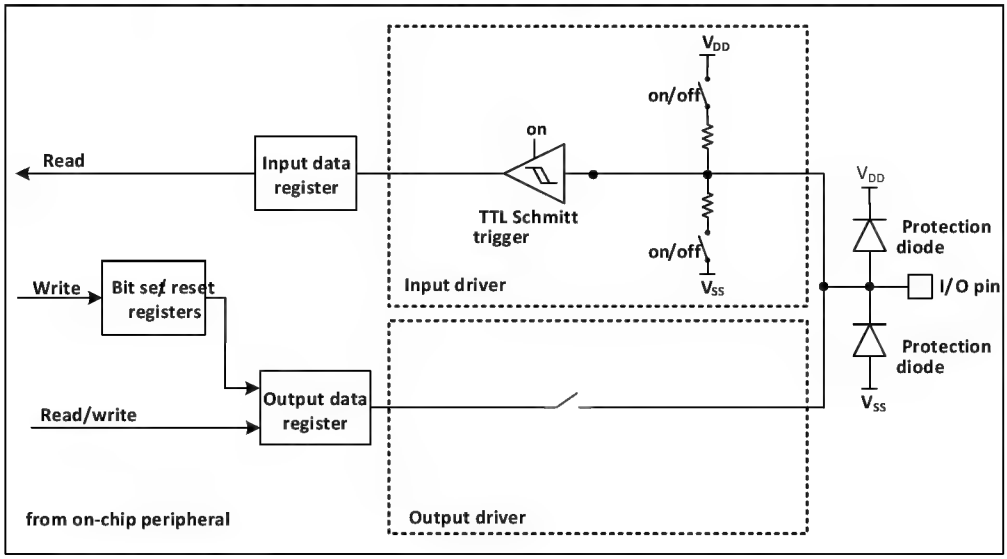
Multiple peripherals may be reused to this pin in the same IO port, so in order to maximize the exertion space of each peripheral, the alternate pin of the peripheral can be remapped and remapped to other pins to avoid the occupied pins in addition to the default alternate pin.

7.2.5 Locking Mechanism

The locking mechanism can lock the configuration of the IO port. After a specific write sequence, the selected IO pin configuration is locked and cannot be changed until the next reset.

7.2.6 Input Configuration

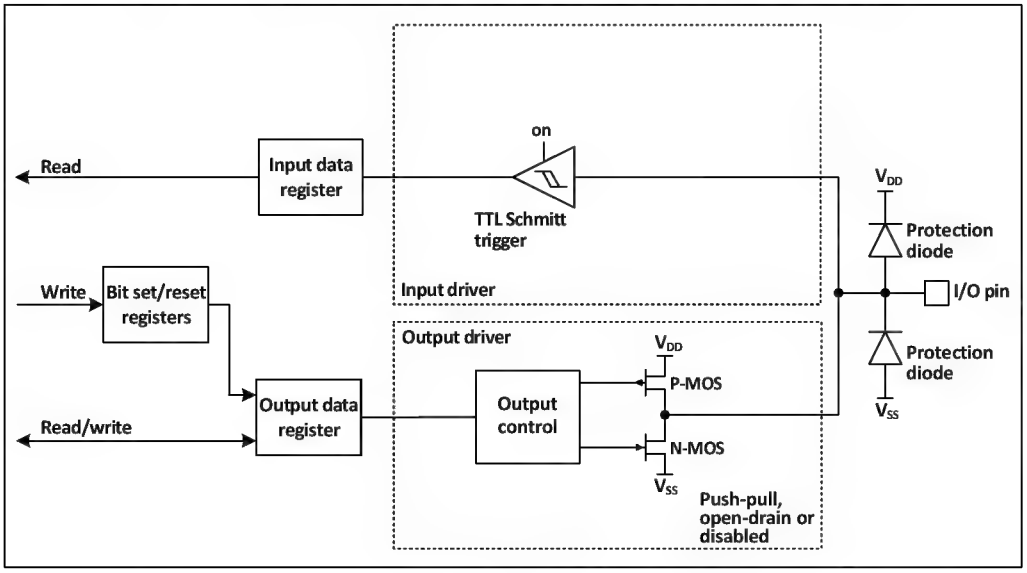
Figure 7-2 GPIO module input configuration structure block diagram



When the IO port is configured in input mode, the output driver is disconnected, the input up and down is optional, and the alternate function and analog input are not connected. The data on each IO port is sampled to the input data register at each HB clock, and the level state of the corresponding pin is obtained by reading the corresponding bit of the input data register.

7.2.7 Output Configuration

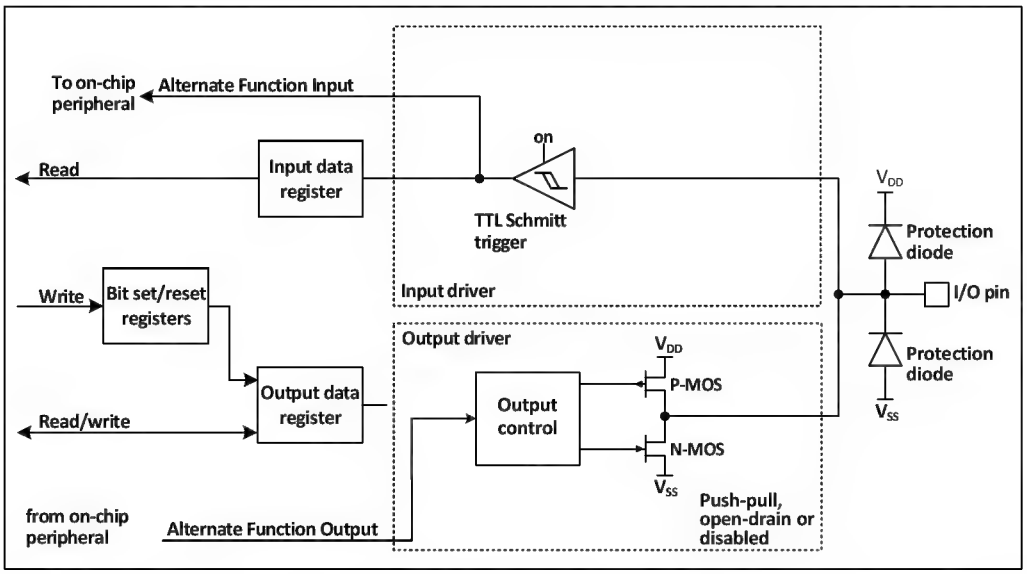
Figure 7-3 GPIO module output configuration structure block diagram



When the IO port is configured in output mode, a pair of MOS in the output driver can be configured in push-pull or open-drain mode as needed, without using the alternate function. The input-driven pull-up resistor is disabled, the TTL Schmitt trigger is activated, and the level that appears on the IO pin will be sampled to the input data register at each HB clock, so reading the input data register will get the IO state, and in push-pull output mode, access to the output data register will get the last written value.

7.2.8 Alternate Function Configuration

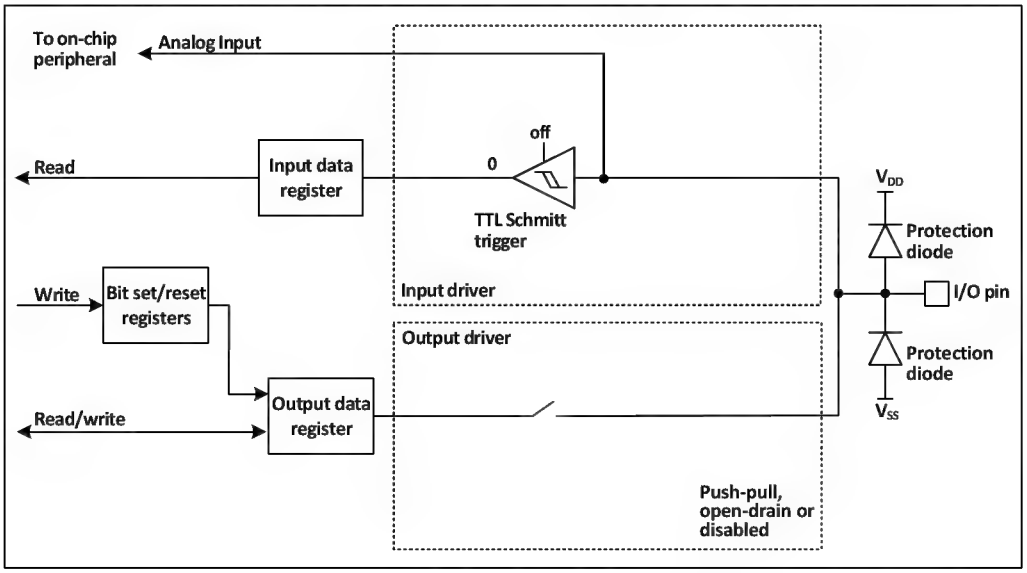
Figure 7-4 The structure of GPIO module when it is multiplexed by other peripherals



When the alternate function is enabled, the output driver is enabled and can be configured in open-drain or push-pull mode as needed, the Schmitt trigger is also turned on, the input and output lines of the alternate function are connected, but the output data register is disconnected, and the level that appears on the IO pin will be sampled to the input data register at each HB clock. In open-drain mode, reading the input data register will get the current state of the IO port. In push-pull mode, reading the output data register will get the last written value.

7.2.9 Analog Input Configuration

Figure 7-5 The configuration structure when the GPIO module is used as an analog input



When the analog input is enabled, the output buffer is disconnected, the input of the Schmitt trigger in the input driver is disabled to prevent consumption on the IO port, the pull-up resistor is prohibited, and the read input data register will always be 0.

7.2.10 GPIO Settings for Peripherals

The following table recommends the corresponding GPIO port configuration for the pins of each peripheral.

Table 7-1 Advanced-control timer (TIM1)

TIM1	Configuration	GPIO configuration
TIM1_CHx	Input capture channel x	Floating input
	Output comparison channel x	Push-pull alternate output
TIM1_CHxN	Complementary output channels x	Push-pull alternate output
TIM1_BKIN	Brake input	Floating input
TIM1_ETR	Externally triggered clock input	Floating input

Table 7-2 General-purpose timer (TIM2)

TIM2 pin	Configuration	GPIO configuration
TIM2_CHx	Input capture channel x	Floating input
	Output comparison channel x	Push-pull alternate output
TIM2_ETR	Externally triggered clock input	Floating input

Table 7-3 Universal synchronous asynchronous serial transceiver (USART)

USART pins	Configuration	GPIO configuration
USARTx_TX	Full-duplex mode	Push-pull alternate outputs
	Half-duplex synchronous mode	Open-drain alternate outputs
USARTx_RX	Full-duplex mode	Floating input or pull-up input
	Half-duplex synchronous mode	Not used
USARTx_RTS	Hardware flow control	Push-pull alternate output
USARTx_CTS	Hardware flow control	Floating input or pull-up input

Table 7-4 Serial peripheral interface (SPI) modules

SPI pins	Configuration	GPIO configuration
SPIx_SCK	Master mode	Push-pull alternate output
	Slave mode	Floating input
SPIx_MOSI	Full-duplex Master mode	Push-pull alternate output
	Full-duplex Slave mode	Floating input or pull-up input
	Simple bi-directional data line/Master mode	Push-pull alternate output
	Simple bi-directional data line/Slave mode	Not used
SPIx_MISO	Full-duplex Master mode	Floating input or pull-up input
	Full-duplex Slave mode	Push-pull alternate output
	Simple bi-directional data line/Master mode	Not used
	Simple bi-directional data line/Slave mode	Push-pull alternate output

SPIx_NSS	Hardware Master or Slave mode	Float, pull-up or pull-down input
	Hardware Master mode/NSS output enable mode	Push-pull alternate output
	Software mode	Not used

Table 7-5 Internal integrated bus (I2C) module

I2C pins	Configuration	GPIO configuration
I2C_SCL	I2C clock	Open-drain alternate output
I2C_SDA	I2C data	Open-drain alternate output

Table 7-6 Analog-to-digital converters (ADC)

ADC pin	GPIO configuration
ADC	Analog input

Table 7-7 Other I/O function settings

Pins	Configuration features	GPIO configuration
MCO	Clock output	Push-pull alternate output
EXTI	External interrupt input	Float, pull-up or pull-down input
OPA	Operational Amplifier Input	Floating input

7.2.11 Alternate Function Remapping GPIO Configuration

7.2.11.1 Timer Alternate Function Remapping

Table 7-8 TIM1 alternate function remapping

Alternate function	TIM1_RM=0000 Default mapping	TIM1_RM=0001 Partial mapping	TIM1_RM=0010 Partial mapping	TIM1_RM=0011 Full mapping	TIM1_RM=0100 Partial mapping	TIM1_RM=0101 Partial mapping	TIM1_RM=0110 Partial mapping	TIM1_RM=0111 Full mapping	TIM1_RM=1000 Full mapping	TIM1_RM=1001 Partial mapping
TIM1_ETR	PC5	PD4	PC5	PC2	PD4	PD4	PD4	PB4	PB4	PB4
TIM1_CH1	PD2	PD2	PC6	PC4	PA3	PA3	PA3	PC4	PC4	PA0
TIM1_CH2	PA1	PA1	PC7	PC7	PB0	PB0	PB0	PC5	PC5	PA1
TIM1_CH3	PC3	PC3	PC0	PC5	PB1	PC3	PB1	PC6	PC6	PA2
TIM1_CH4	PC4	PC4	PD3	PD4	PD1	PD1	PB2	PC7	PC7	PA3
TIM1_BKIN	PC2	PC2	PC1	PC1	PB3	PB3	PA7	PB2	PB2	PB2
TIM1_CH1N	PD0	PD0	PC3	PC3	PA0	PA0	PA0	PC0	PA3	PC0
TIM1_CH2N	PA2	PA2	PC4	PD2	PA2	PA2	PA2	PC1	PB0	PC1
TIM1_CH3N	PD1	PD1	PD1	PC6	PD0	PD0	PD0	PC2	PB1	PC2

Note: For the mapping function of TIM1_CH1 in the table, the condition is TIM1_1_RM=0. When TIM1_1_RM=1, TIM1_CHI maps to LSI (For LSI calibration)

Table 7-9-1 TIM2 alternate function remapping (CH32V002/004/005/006 only)

Alternate function	TIM2_RM=000 Default	TIM2_RM=001 Partial	TIM2_RM=010 Full	TIM2_RM=011 Full	TIM2_RM=100 Full	TIM2_RM=101 Full	TIM2_RM=110 Full	TIM2_RM=111 Full

	mapping	mapping	mapping	mapping	mapping	mapping	mapping	mapping
TIM2_ETR	PD4	PC1	PC5	PC1	PC0	PA0	PB1	PD3
TIM2_CH1	PD4	PC1	PC5	PC1	PC0	PA0	PB1	PD3
TIM2_CH2	PD3	PD3	PC2	PC7	PC1	PA1	PA1	PD4
TIM2_CH3	PC0	PC0	PD2	PD6	PC3	PA2	PA2	PA2
TIM2_CH4	PD7	PD7	PC1	PD5	PB6	PA3	PA3	PA3

Table 7-9-2 TIM2 alternate function remapping (CH32V007/CH32M007 only)

Alternate function	TIM2_RM=000 Default mapping	TIM2_RM=001 Partial mapping	TIM2_RM=010 Full mapping	TIM2_RM=011 Full mapping	TIM2_RM=100 Full mapping	TIM2_RM=101 Full mapping	TIM2_RM=110 Full mapping	TIM2_RM=111 Full mapping
TIM2_ETR	PD4	PC1	PC5	PC1	PC0	PA0	PB1	PD3
TIM2_CH1	PD4	PC1	PC5	PC1	PC0	PA0	PB1	PD3
TIM2_CH2	PD3	PD3	PB3	PC7	PC1	PA1	PA1	PD4
TIM2_CH3	PC0	PC0	PD2	PD6	PC3	PA2	PA2	PA2
TIM2_CH4	PD7	PD7	PC1	PD5	PB6	PA3	PA3	PA3

7.2.11.2 USART Alternate Function Remapping

Table 7-10 USART1 alternate function remapping

Alternate function	USART1_RM=0000 Default mapping	USART1_RM=0001 Remapping	USART1_RM=0010 Remapping	USART1_RM=0011 Remapping	USART1_RM=0100 Remapping	USART1_RM=0101 Remapping	USART1_RM=0110 Remapping	USART1_RM=0111 Remapping	USART1_RM=1000 Remapping	USART1_RM=1001 Remapping
USART1_TX	PD5	PD6	PD0	PC0	PD1	PB3	PC5	PB5	PA0	PA0
USART1_RX	PD6	PD5	PD1	PC1	PB3	PD1	PC6	PB6	PA1	PC4
USART1_CTS	PD3	PC6	PC3	PC6	PD7	PD7	PC7	PC7	PD2	PD5
USART1_RTS	PC2	PC7	PC2	PC7	PA5	PA5	PB4	PB4	PD3	PD4

Table 7-11 USART2 alternate function remapping

Alternate function	USART2_RM=000 Default mapping	USART2_RM=001 Remapping	USART2_RM=010 Remapping	USART2_RM=011 Remapping	USART2_RM=100 Remapping	USART2_RM=101 Remapping	USART2_RM=110 Remapping
USART2_TX	PA7	PA4	PA2	PD2	PB0	PC4	PA6
USART2_RX	PB3	PA5	PA3	PD3	PB1	PD1	PA5
USART2_CTS	PA4	PA7	PA0	PA0	PB6	PA4	PA7

USART2_RTS	PA5	PB3	PA1	PA1	PA1	PA1	PB3

7.2.11.3 SPI Alternate Function Remapping

Table 7-12 SPI alternate function remapping

Alternate function	SPI_RM=000 Default mapping	SPI_RM=001 Remapping	SPI_RM=010 Remapping	SPI_RM=011 Remapping	SPI_RM=100 Remapping	SPI_RM=101 Remapping	SPI_RM=110 Remapping
SPI_NSS	PC1	PC0	PC4	PB0	PD3	PC1	PC4
SPI_SCK	PC5	PC5	PD2	PB1	PD4	PA1	PB5
SPI_MISO	PC7	PC7	PB3	PB2	PD5	PB5	PC7
SPI_MOSI	PC6	PC6	PD3	PC0	PD6	PA2	PB4

7.2.11.4 I2C Alternate Function Remapping

Table 7-13-1 I2C alternate function remapping (CH32V002/004/005/006 only)

Alternate function	I2C_RM=000 Default mapping	I2C_RM=001 Remapping	I2C_RM=010 Remapping	I2C_RM=011 Remapping	I2C_RM=1xx Remapping
I2C_SCL	PC2	PD1	PC5	PB5	PB3
I2C_SDA	PC1	PD0	PC6	PB6	PD1

Table 7-13-2 I2C alternate function remapping (CH32V007/CH32M007 only)

Alternate function	I2C_RM=000 Default mapping	I2C_RM=001 Remapping	I2C_RM=010 Remapping	I2C_RM=011 Remapping	I2C_RM=1xx Remapping
I2C_SCL	PC2	PD1	PC5	PB5	PB3
I2C_SDA	PC1	PD0	PC4	PB6	PD1

7.2.11.5 ADC Alternate Function Remapping

Table 7-14 ADC external trigger injection conversion alternate function remapping

Alternate function	ADC_ETRGINJ_RM=0 Default mapping	ADC_ETRGINJ_RM=1 Remapping
ADC external trigger injection conversion	ADC external trigger injection conversion connect to PD1	ADC external trigger injection conversion connect to PA2

Table 7-15 ADC external trigger rule conversion alternate function remapping

Alternate function	ADC_ETRGREG_RM=0 Default mapping	ADC_ETRGREG_RM=1 Remapping
ADC external trigger rule conversion	ADC external trigger rule conversion connect to PD3	ADC external trigger rule conversion connect to PC2

7.3 Register Description

7.3.1 GPIO Register Description

Unless otherwise specified, the registers of the GPIO must be operated as words (Operate these registers with 32 bits).

Table 7-16 GPIO-related registers list

Name	Access address	Description	Reset value
R32_GPIOA_CFGLR	0x40010800	PA Port Configuration Register Low	0x44444444
R32_GPIOB_CFGLR	0x40010C00	PB Port Configuration Register Low	0x44444444
R32_GPIOC_CFGLR	0x40011000	PC Port Configuration Register Low	0x44444444
R32_GPIOD_CFGLR	0x40011400	PD Port Configuration Register Low	0x44444444
R32_GPIOA_INDR	0x40010808	PA Port Input Data Register	0x000000XX
R32_GPIOB_INDR	0x40010C08	PB Port Input Data Register	0x000000XX
R32_GPIOC_INDR	0x40011008	PC Port Input Data Register	0x000000XX
R32_GPIOD_INDR	0x40011408	PD Port Input Data Register	0x000000XX
R32_GPIOA_OUTDR	0x4001080C	PA Port Output Data Register	0x00000000
R32_GPIOB_OUTDR	0x40010C0C	PB Port Output Data Register	0x00000000
R32_GPIOC_OUTDR	0x4001100C	PC Port Output Data Register	0x00000000
R32_GPIOD_OUTDR	0x4001140C	PD Port Output Data Register	0x00000000
R32_GPIOA_BSHR	0x40010810	PA Port Set/Reset Register	0x00000000
R32_GPIOB_BSHR	0x40010C10	PB Port Set/Reset Register	0x00000000
R32_GPIOC_BSHR	0x40011010	PC Port Set/Reset Register	0x00000000
R32_GPIOD_BSHR	0x40011410	PD Port Set/Reset Register	0x00000000
R32_GPIOA_BCR	0x40010814	PA Port Reset Register	0x00000000
R32_GPIOB_BCR	0x40010C14	PB Port Reset Register	0x00000000
R32_GPIOC_BCR	0x40011014	PC Port Reset Register	0x00000000
R32_GPIOD_BCR	0x40011414	PD Port Reset Register	0x00000000
R32_GPIOA_LCKR	0x40010818	PA Port Lock Configuration Register	0x00000000
R32_GPIOB_LCKR	0x40010C18	PB Port Lock Configuration Register	0x00000000
R32_GPIOC_LCKR	0x40011018	PC Port Lock Configuration Register	0x00000000
R32_GPIOD_LCKR	0x40011418	PD Port Lock Configuration Register	0x00000000

7.3.1.1 Port Configuration Register Low (GPIOx_CFGLR) (x=A/B/C/D)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNF7[1:0]		Reserved	MODE7	CNF6[1:0]		Reserved	MODE6	CNF5[1:0]		Reserved	MODE5	CNF4[1:0]		Reserved	MODE4
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNF3[1:0]		Reserved	MODE3	CNF2[1:0]		Reserved	MODE2	CNF1[1:0]		Reserved	MODE1	CNF0[1:0]		Reserved	MODE0

Bit	Name	Access	Description	Reset value

[31:30] [27:26] [23:22] [19:18] [15:14] [11:10] [7:6] [3:2]	CNFy[1:0]	RW	(y=0-7), the configuration bits of port x, through which the corresponding ports are configured. When entering a mode (MODE=00b): 00: Analog input mode; 01: Floating input mode; 10: With pull-up mode; 11: Reserved. In output mode (MODE>00b): 00: General push-pull output mode; 01: General open-drain output mode; 10: Alternate function push-pull output mode; 11: Alternate function open-drain output mode.	01b
29 25 21 17 13 9 5 1	Reserved	RO	Reserved	0
28 24 20 16 12 8 4 0	MODEy	RW	(y=0-7), port x mode selection, configure the corresponding port through these bits. 1: Output mode, maximum speed 30MHz; 0: Input mode.	0

7.3.1.2 Port Input Data Register (GPIOx_INDR) (x=A/B/C/D)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	IDRy	RO	(y=0-7), the port inputs data. These bits are read-only and can only be read in 16-bit form. The value read out is the high and low state of the corresponding bit.	X

7.3.1.3 Port Output Data Register (GPIOx_OUTDR) (x=A/B/C/D)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	ODRy	RW	For output mode: (y=0-7), data output from the port. This data can only be manipulated in 16-bit form. The IO port outputs the values of these registers. For with pull-up input mode: 1: Pull-up input; 0: Pull-down input.	0

7.3.1.4 Port Reset/Set Register (GPIOx_BSHR) (x=A/B/C/D)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	BRy	WO	(y=0-7), the corresponding OUTDR bits are cleared for these bits, and writing 0 has no effect. These bits can only be accessed in the form of 16 bits. If both the BR and BS bits are set, the BS bit works.	0
[15:8]	Reserved	RO	Reserved	0
[7:0]	BSy	WO	(y=0-7), writing 0 does not affect the corresponding OUTDR position bits for these position bits. These bits can only be accessed in the form of 16 bits. If both the BR and BS bits are set, the BS bit works.	0

7.3.1.5 Port Reset Register (GPIOx_BCR) (x=A/B/C/D)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	BRy	WO	(y=0-7), the corresponding OUTDR bits are cleared for these bits, and writing 0 has no effect. These bits can only be accessed in the form of 16 bits.	0

7.3.1.6 Port Configuration Lock Register (GPIOx_LCKR) (x=A/B/C/D)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LCKK
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LCKK	RW	Lock key, which can be locked by writing a specific sequence, but it can be read out at any time. When it reads 0, it means that the lock is not in effect, and read 1 means that the lock is in effect. The write sequence of the lock key is: write 1-write 0-write 1-read 0-read 1, the last step is not necessary, but can be used to confirm that the lock key has been activated. Any error when writing a sequence does not activate a lock, and the value of LCK [7:0] cannot be changed when writing a sequence. After the lock is in effect, the configuration of the port can be changed only after the next reset.	0
[15:8]	Reserved	RO	Reserved	0
[7:0]	LCKy	RW	(y=0-7), where 1 indicates that the configuration of the corresponding port is locked. These bits can only be changed before the LCKK is locked. Locked configurations refer to the configuration registers GPIOx_CFGLR and GPIOx_CFGHR.	0

Note: When the LOCK sequence is executed on the corresponding port bit, the configuration of the port bit will not be changed until the next system reset.

7.3.2 AFIO Register Description

Unless otherwise specified, AFIO registers must be operated as words (operate these registers with 32 bits).

Table 7-17 List of AFIO-related registers

Name	Access address	Description	Reset value
R32_AFIO_EXTICR	0x40010008	External Interrupt Configuration Register 1	0x00000000
R32_AFIO_PCFR1	0x4001000C	Remapping Register 1	0x00000000

7.3.2.1 External Interrupt Configuration Register 1 (AFIO_EXTICR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[1:0]		EXTI6[1:0]		EXTI5[1:0]		EXTI4[1:0]		EXTI3[1:0]		EXTI2[1:0]		EXTI1[1:0]		EXTI0[1:0]	

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:14] [13:12] [11:10] [9:8] [7:6] [5:4] [3:2] [1:0]	EXTIx[1:0]	RW	(x=0-7), external interrupt input pin configuration bit. It is used to determine which port pin the external interrupt pin is mapped to. 00: The x-pin of the PA pin; 01: The x-pin of the PB pin; 10: The x-pin of the PC pin; 11: The x-pin of the PD pin.	0

7.3.2.2 Remap Register 1 (AFIO_PCFR1)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					SWCFG[2:0]			Reserved	USART2_RM[2:0]			ADC_ETRGREG_RM	ADC_ETRGINJ_RM	PA1PA2_RM	TIM2_RM[2]
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIM2_RM[1:0]		TIM1_RM[3:0]				USART1_RM[3:0]				I2C1_RM[2:0]			SPI1_RM[2:0]		

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
[26:24]	SWCFG[2:0]	RW	These bits are used to configure the SW function and the IO port of the tracking function. SWD (SDI) is the debug interface that accesses the core.	0

			The system is always used as a SWD port when it is reset. 0xx: Enable SWD (SDI); 100: Disable SWD (SDI), as GPIO function; Others: Invalid.	
23	Reserved	RO	Reserved	0
[22:20]	USART2_RM[2:0]	RW	USART2 remapping bit. 000: Default mapping (TX/PA7, RX/PB3, CTS/PA4, RTS/PA5); 001: Remapping (TX/PA4, RX/PA5, CTS/PA7, RTS/PB3); 010: Remapping (TX/PA2, RX/PA3, CTS/PA0, RTS/PA1); 011: Remapping (TX/PD2, RX/PD3, CTS/PA0, RTS/PA1); 100: Remapping (TX/PB0, RX/PB1, CTS/PB6, RTS/PA1); 101: Remapping (TX/PC4, RX/PD1, CTS/PA4, RTS/PA11); 110: Remapping (TX/PA6, RX/PA5, CTS/PA7, RTS/PB3); Others: Reserved.	0
19	ADC_ETRGREG_RM	RW	Remap bit of ADC external triggered rule conversion. 1: ADC external triggered rule conversion is connected to PC2; 0: ADC external triggered rule conversion is connected to PD3.	0
18	ADC_ETRGINJ_RM	RW	Remap bit of ADC external triggered injection conversion。 1: ADC external triggered injection conversion is connected to PA2; 0: ADC external triggered injection conversion is connected to PD1.	0
17	PA1PA2_RM	RW	Pin PA1&PA2 remap bit, which can be read and written by the user. It controls whether the function of PA1 and PA2 is normal (1 is required to connect external crystal pins). 1: The pin has no function. 0: Pins are used as GPIO and reuse functions. *Note: PA1 and PA2 cannot be used as GPIO function when they are used as crystal pins.*	0
[16:14]	TIM2_RM[2:0]	RW	The remap bit of timer 2. These bits can be read and written by the user. It controls the mapping of	0

			channels 1 to 4 of timer 2 and external triggers (ETR) on the GPIO port. 000: Default mapping (CH1/ETR/PD4, CH2/PD3, CH3/CH1N/PC0, CH4/CH2N/PD7); 001: Partial mapping (CH1/ETR/PC1, CH2/PD3, CH3/PC0, CH4/CH2N/PD7); 010 (Only for CH32V002/004/005/006): Full mapping (CH1/ETR/PC5, CH2/PC2, CH3/CH1N/PD2, CH4/CH2N/PC1); 010 (Only for CH32V007/CH32M007): Full mapping (CH1/ETR/PC5, CH2/PB3, CH3/CH1N/PD2, CH4/CH2N/PC1); 011: Full mapping (CH1/ETR/PC1, CH2/PC7, CH3/CH1N/PD6, CH4/CH2N/PD5); 100: Full mapping (CH1/ETR/PC0, CH2/PC1, CH3/PC3, CH4/CH2N/PB6); 101: Full mapping (CH1/ETR/PA0, CH2/PA1, CH3/CH1N/PA2, CH4/CH2N/PA3); 110: Full mapping (CH1/ETR/PB1, CH2/PA1, CH3/CH1N/PA2, CH4/CH2N/PA3); 111: Full mapping (CH1/ETR/PD3, CH2/PD4, CH3/CH1N/PA2, CH4/CH2N/PA3).	
[13:10]	TIM1_RM[3:0]	RW	The remap bit of timer 1. These bits can be read and written by the user. It controls the mapping of channels 1 to 4, 1N to 3N, external trigger (ETR) and brake input (BKIN) of timer 1 on the GPIO port. 0000: Default mapping (ETR/PC5, CH1/PD2, CH2/PA1, CH3/PC3, CH4/PC4, BKIN/PC2, CH1N/PD0, CH2N/PA2, CH3N/PD1); 0001: Partial mapping (ETR/PD4, CH1/PD2, CH2/PA1, CH3/PC3, CH4/PC4, BKIN/PC2, CH1N/PD0, CH2N/PA2, CH3N/PD1); 0010: Partial mapping (ETR/PC5, CH1/PC6, CH2/PC7, CH3/PC0, CH4/PD3, BKIN/PC1, CH1N/PC3, CH2N/PC4, CH3N/PD1); 0011: Full mapping (ETR/PC2, CH1/PC4, CH2/PC7, CH3/PC5, CH4/PD4, BKIN/PC1, CH1N/PC3, CH2N/PD2, CH3N/PC6); 0100: Partial mapping (ETR/PD4, CH1/PA3, CH2/PB0, CH3/PB1, CH4/PD1, BKIN/PB3, CH1N/PA0, CH2N/PA2, CH3N/PD0); 0101: Partial mapping (ETR/PD4, CH1/PA3, CH2/PB0, CH3/PC3, CH4/PD1, BKIN/PB3, CH1N/PA0, CH2N/PA2, CH3N/PD0);	0

			0110: Partial mapping (ETR/PD4, CH1/PA3, CH2/PB0, CH3/PB1, CH4/PB2, BKIN/PA7, CH1N/PA0, CH2N/PA2, CH3N/PD0); 0111: Full mapping (ETR/PB4, CH1/PC4, CH2/PC5, CH3/PC6, CH4/PC7, BKIN/PB2, CH1N/PC0, CH2N/PC1, CH3N/PC2); 1000: Full mapping (ETR/PB4, CH1/PC4, CH2/PC5, CH3/PC6, CH4/PC7, BKIN/PB2, CH1N/PA3, CH2N/PB0, CH3N/PB1); 1001: Partial mapping (ETR/PB4, CH1/PA0, CH2/PA1, CH3/PA2, CH4/PA3, BKIN/PB2, CH1N/PC0, CH2N/PC1, CH3N/PC2); 11xx: Control the input of channel 1 of timer 1 to select the internal LSI clock; Others: Reserved.	
[9:6]	USART1_RM[3:0]	RW	USART1 remap bit. 0000: Default mapping (TX/PD5, RX/PD6, CTS/PD3, RTS/PC2); 0001: Remapping (TX/PD6, RX/PD5, CTS/PC6, RTS/PC7); 0010: Remapping (TX/PD0, RX/PD1, CTS/PC3, RTS/PC2); 0011: Remapping (TX/PC0, RX/PC1, CTS/PC6, RTS/PC7); 0100: Remapping (TX/PD1, RX/PB3, CTS/PD7, RTS/PA5); 0101: Remapping (TX/PB3, RX/PD1, CTS/PD7, RTS/PA5); 0110: Remapping (TX/PC5, RX/PC6, CTS/PC7, RTS/PB4); 0111: Remapping (TX/PB5, RX/PB6, CTS/PC7, RTS/PB4); 1000: Remapping (TX/PA0, RX/PA1, CTS/PD2, RTS/PD3); 1001: Remapping (TX/PA0, RX/PC4, CTS/PD5, RTS/PD4); Others: Reserved.	0
[5:3]	I2C1_RM[2:0]	RW	I2C1 remap bit. 000: Default mapping (SCL/PC2, SDA/PC1); 001: Remapping (SCL/PD1, SDA/PD0); 010 (Only for CH32V002/004/005/006): Remapping (SCL/PC5, SDA/PC6); 010 (Only for CH32V007/CH32M007): Remapping (SCL/PC5, SDA/PC4);	0

			011: Remapping (SCL/PB5, SDA/PB6); 1xx: Remapping (SCL/PB3, SDA/PD1).	
[2:0]	SPI1_RM[2:0]	RW	Remapping of SPI1. This bit can be read and written by the user. It controls the mapping of SPI1's NSS, SCK, MISO and MOSI multiplexing functions on GPIO ports. 000: Default mapping (NSS/PC1, SCK/PC5, MISO/PC7, MOSI/PC6); 001: Remapping (NSS/PC0, SCK/PC5, MISO/PC7, MOSI/PC6); 010: Remapping (NSS/PC4, SCK/PD2, MISO/PB3, MOSI/PD3); 011: Remapping (NSS/PB0, SCK/PB1, MISO/PB2, MOSI/PC0); 100: Remapping (NSS/PD3, SCK/PD4, MISO/PD5, MOSI/PD6); 101: Remapping (NSS/PC1, SCK/PA1, MISO/PB5, MOSI/PA2); 110: Remapping (NSS/PC4, SCK/PB5, MISO/PC7, MOSI/PB4); 111: Reserved.	0

Chapter 8 Direct Memory Access Control (DMA)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

Direct memory access controller (DMA) provides a high-speed data transfer mode between peripherals and memory or between memory and memory. Without CPU intervention, data can be moved quickly through DMA to save CPU resources for other operations.

Each channel of the DMA controller is dedicated to managing requests for memory access from one or more peripherals. There is also an arbitrator to coordinate priorities between channels.

8.1 Main Features

- Multiple independently configurable channels
- Each channel is directly connected to dedicated hardware DMA requests and supports software triggering.
- Buffer management supporting loops
- The priority of the request between multiple channels can be set by software programming (Highest, high, medium and low). When the priority setting is equal, it is determined by the channel number (The lower the channel number, the higher the priority).
- Support peripherals to memory, memory to peripherals, and memory to memory transfer
- Flash memory, SRAM, SRAM of peripherals and HB peripherals can be used as sources and destinations for access
- Number of programmable data transfer bytes: Up to 65535

8.2 Function Description

8.2.1 DMA Channel Processing

1) Arbitration priority

DMA requests generated by multiple independent channels are logically or structurally input to the DMA controller, and currently only one channel request is answered. The arbitrator within the module selects the access of the peripheral / memory to be initiated according to the priority of the channel request.

In software management, applications can independently configure priority levels for each channel, including the highest, high, medium and low levels, by setting the PL [1:0] bit of the DMA_CFGRx register. When the software settings between the channels are the same, the modules will be selected according to the fixed hardware priority, and the lower channel number will have higher priority than the higher one.

2) DMA configuration

When the DMA controller receives a request signal, it accesses the requesting peripheral or memory to establish the peripheral or data transfer between the memory and the memory. It mainly includes the following three steps:

1) Data is taken from the memory address indicated by the peripheral data register or the current peripheral / memory address register, and the starting address of the first transmission is the peripheral base address or memory address specified by the DMA_PADDRx or DMA_MADDRx register.
2) The data is stored in the peripheral data register or the memory address indicated by the current peripheral / memory address register, and the starting address of the first transmission is the peripheral base address or memory address specified in the DMA_PADDRx or DMA_MADDRx register.
3) Performs a decrement operation of values in a DMA_CNTRx register that indicates the number of operations

currently outstanding for transfer.

Each channel includes three DMA data transfer methods:

- Peripheral to memory (MEM2MEM=0, DIR=0)
- Memory to peripheral (MEM2MEM=0, DIR=1)
- Memory to memory (MEM2MEM=1)

Note: The memory-to-memory mode does not require peripheral request signals. After configuring this mode (MEM2MEM=1), the channel can be opened (EN=1) to start data transmission. Circular mode is not supported in this way.

The configuration process is as follows:

1) The first address of the peripheral register or the memory data address in the memory-to-memory mode (MEM2MEM=1) is set in the DMA_PADDRx register. When an DMA request occurs, this address will be the source or destination address of the data transfer.
2) Set the memory data address in the DMA_MADDRx register. When a DMA request occurs, the transmitted data will be read from or written to this address.
3) Set the amount of data to be transferred in the DMA_CNTRx register. After each data transmission, this value decreases.
4) Set the priority of the channel in the PL [1:0] bit of the DMA_CFGRx register.
5) The direction of data transmission, loop mode, incremental mode of peripherals and memory, data width of peripherals and memory, more than half of transmission, transmission completion, and transmission error interrupt enable bit are set in the DMA_CFGRx register.
6) Set the EN bit of the DMA_CFGRx register to start channel x.

Note: The control bits in the DMA_PADDRx/DMA_MADDRx/DMA_CNTRx register and the data transfer direction (DIR), loop mode (location), peripheral and memory incremental mode (MINC/PINC) in the DMA_CFGRx register can be configured to write only when the DMA channel is turned off.

3) Circular mode

Set the CIRC location 1 of the DMA_CFGRx register to enable the circular mode function of channel data transfer. In circular mode, when the number of data transfers becomes 0, the contents of the DMA_CNTRx register are automatically reloaded to their initial values, the internal peripherals and memory address registers are also reloaded to the initial address values set by the DMA_PADDRx and DMA_MADDRx registers, and the DMA operation continues until the channel is closed or DMA mode is turned off.

4) DMA processing status

- Transfer more than half: corresponding to the HTIFx bit hardware setting in the DMA_INTFR register. When the number of transmission bytes of DMA is reduced to less than half of the initial setting value, more than half of the DMA transfer flag will be generated, and if HTIE is set in the DMA_CFGRx register, an interrupt will occur. The hardware uses this flag to remind the application that it can prepare for a new round of data transfer.
- Transfer completed: corresponding to the TCIFx bit hardware setting in the DMA_INTFR register. When the number of transfer bytes of DMA is reduced to 0, the DMA transfer completion flag will be generated, and if TCIE is set in the DMA_CFGRx register, an interrupt will occur.
- Transmission error: corresponding to the TEIFx bit hardware setting in the DMA_INTFR register. Reading and writing a reserved address area will result in a DMA transmission error. At the same time, the module

hardware will automatically clear the EN bits of the DMA_CFGRx register corresponding to the channel where the error occurred, and the channel is closed. If TEIE is set in the DMA_CFGRx register, an interrupt will occur.

When querying the status of the DMA channel, the application can first access the GIFx bit of the DMA_INTFR register, determine which channel has the DMA event, and then deal with the specific DMA event content of the channel.

8.2.2 Programmable Total Data Transfer Size/Data Bit Width/Alignment

The total amount of data transmitted by DMA in one round of each channel is programmable, up to 65535 times. The DMA_CNTRx register indicates the number of bytes to be transferred. In EN=0, the setting value is written, and after the EN=1 opens the DMA transmission channel, the register becomes read-only, and the value decreases after each transfer.

The transmission data values of peripherals and memory support the function of automatic increment of address pointer, and the pointer increment is programmable. The first transmitted data address they access is stored in the DMA_PADDRx and DMA_MADDRx registers. By setting the PINC bit or MINC location 1 of the DMA_CFGRx register, you can turn on the peripheral address self-increment mode or the memory address self-increment mode, respectively. PSIZE [1:0] sets the peripheral address to take data size and address self-increase, and MSIZE [1:0] sets the memory address to take data size and address self-increase. There are 3 options: 8-bit, 16-bit, 32-bit. The specific data transfer methods are as follows:

Table 8-1 DMA transfer with different data bit widths (PINC=MINC=1)

Source bit width	Objectives bit width	Transmission number	Source: address/data	Target: address/data	Transfer operations
8	8	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/B0 0x01/B1 0x02/B2 0x03/B3	● The incremental address of the source side is aligned with the data bit width set by the source side, and the value is equal to the data bit width of the source side. ● The increment of the destination address is aligned with the bit width of the target setting data, and the value is equal to the bit width of the target data. ● DMA transfer to the target end of the data according to the principle: the data size is not enough high-order fill 0, data size overflow high-order removal ● Data storage mode:
8	16	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/00B0 0x02/00B1 0x04/00B2 0x06/00B3	
8	32	4	0x00/B0 0x01/B1 0x02/B2 0x03/B3	0x00/000000B0 0x04/000000B1 0x08/000000B2 0x0C/000000B3	
16	8	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B0 0x01/B2 0x02/B4 0x03/B6	
16	16	4	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	0x00/B1B0 0x02/B3B2 0x04/B5B4 0x06/B7B6	
16	32	4	0x00/B1B0 0x02/B3B2	0x00/0000B1B0 0x04/0000B3B2	

			0x04/B5B4 0x06/B7B6	0x08/0000B5B4 0x0C/0000B7B6	small end mode, low address stores low bytes, high address stores high bytes
32	8	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B0 0x01/B4 0x02/B8 0x03/BC	
32	16	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B1B0 0x02/B5B4 0x04/B9B8 0x06/BDBC	
32	32	4	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	0x00/B3B2B1B0 0x04/B7B6B5B4 0x08/BBBAB9B8 0x0C/BFBEBDBC	

8.2.3 DMA Request Mapping

The DMA controller provides 7 channels, and each channel corresponds to multiple peripheral requests. By setting the corresponding DMA control bit in the corresponding peripheral register, the DMA function of each peripheral can be turned on or off independently. The specific correspondence is as follows.

Figure 8-1 DMA1 request image

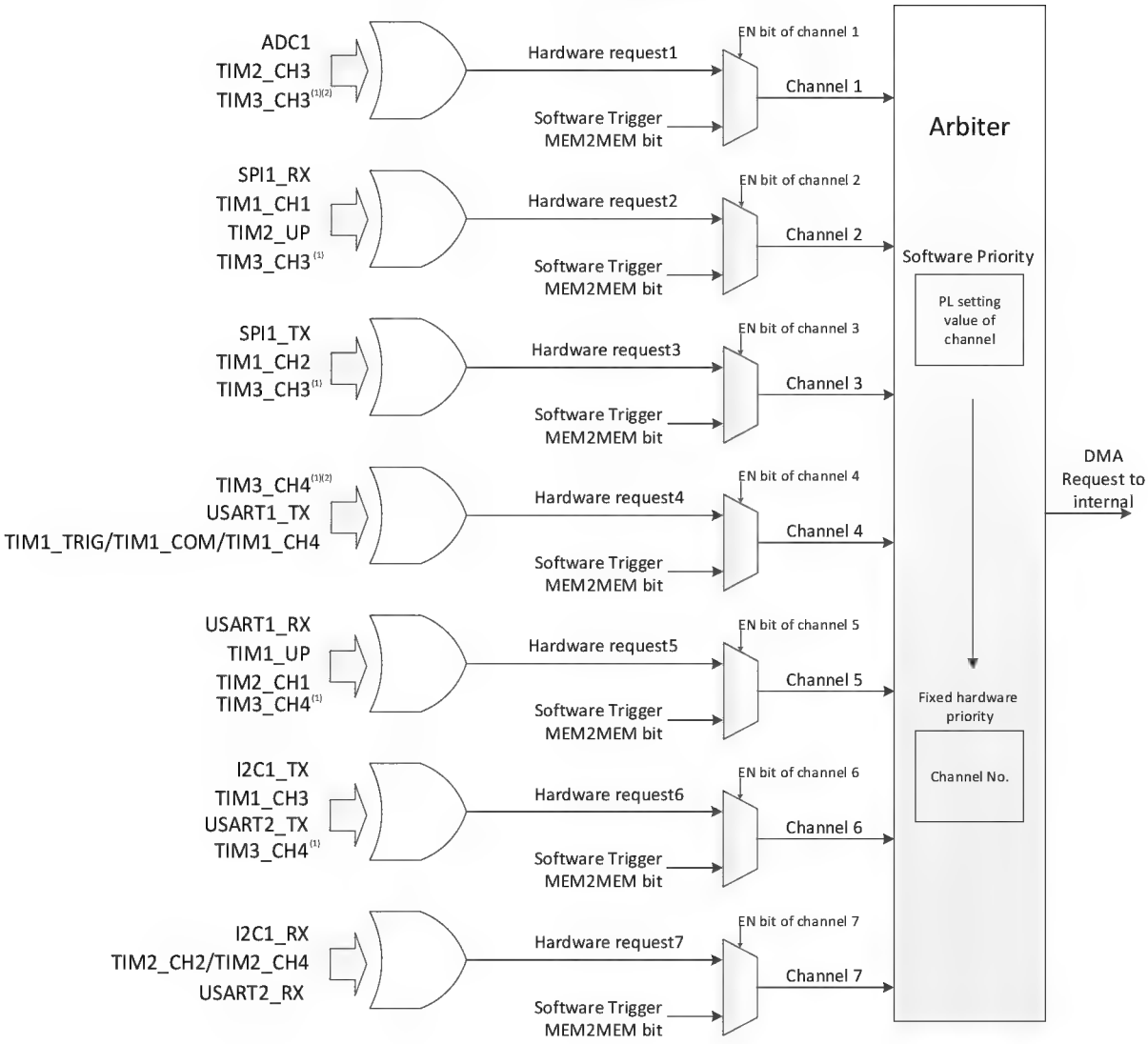


Table 8-2 DMA1 peripheral mapping table for each channel

Peripheral	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6	Channel 7
ADC	ADC						
SPI1		SPI1_RX	SPI1_TX				
USART1				USART1_TX	USART1_RX		
USART2						USART2_TX	USART2_RX
I2C1						I2C1_TX	I2C1_RX
TIM1		TIM1_CH1	TIM1_CH2	TIM1_CH4 TIM1_TRIG TIM1_COM	TIM1_UP	TIM1_CH3	
TIM2	TIM2_CH3	TIM2_UP			TIM2_CH1		TIM2_CH2 TIM2_CH4
TIM3	TIM3_CH3(1)(2)	TIM3_CH3(1)	TIM3_CH3(1)	TIM3_CH4(1)(2)	TIM3_CH4(1)	TIM3_CH4(1)	

Note:

(1) The TIM3 of CH32M007 generates a CH3 event, which can trigger channels 1, 2 and 3 of DMA1, and TIM3 generates an CH3 event that can trigger channels 4, 5 and 6 of DMA1.

(2) TIM3 of CH32V006 and CH32V007 generates one CH3 event, which can only trigger channel 1 of DMA1; TIM3 generates one CH4 event, which can only trigger channel 4 of DMA1.

8.3 Register Description

Table 8-3 DMA-related registers list

Name	Access address	Description	Reset value
R32_DMA_INTFR	0x40020000	DMA interrupt status register	0x00000000
R32_DMA_INTFCR	0x40020004	DMA interrupt flag clear register	0x00000000
R32_DMA_CFGR1	0x40020008	DMA channel 1 configuration register	0x00000000
R32_DMA_CNTR1	0x4002000C	DMA channel 1 transfer data number register	0x00000000
R32_DMA_PADDR1	0x40020010	DMA channel 1 peripheral address register	0x00000000
R32_DMA_MADDR1	0x40020014	DMA channel 1 memory address register	0x00000000
R32_DMA_CFGR2	0x4002001C	DMA channel 2 configuration register	0x00000000
R32_DMA_CNTR2	0x40020020	DMA channel 2 transfer data number register	0x00000000
R32_DMA_PADDR2	0x40020024	DMA channel 2 peripheral address register	0x00000000
R32_DMA_MADDR2	0x40020028	DMA channel 2 memory address register	0x00000000
R32_DMA_CFGR3	0x40020030	DMA channel 3 configuration register	0x00000000
R32_DMA_CNTR3	0x40020034	DMA channel 3 transfer data number register	0x00000000
R32_DMA_PADDR3	0x40020038	DMA channel 3 peripheral address register	0x00000000
R32_DMA_MADDR3	0x4002003C	DMA channel 3 memory address register	0x00000000
R32_DMA_CFGR4	0x40020044	DMA channel 4 configuration register	0x00000000
R32_DMA_CNTR4	0x40020048	DMA channel 4 transfer data number register	0x00000000
R32_DMA_PADDR4	0x4002004C	DMA channel 4 peripheral address register	0x00000000
R32_DMA_MADDR4	0x40020050	DMA channel 4 memory address register	0x00000000
R32_DMA_CFGR5	0x40020058	DMA channel 5 configuration register	0x00000000
R32_DMA_CNTR5	0x4002005C	DMA channel 5 transfer data number register	0x00000000
R32_DMA_PADDR5	0x40020060	DMA channel 5 peripheral address register	0x00000000
R32_DMA_MADDR5	0x40020064	DMA channel 5 memory address register	0x00000000
R32_DMA_CFGR6	0x4002006C	DMA channel 6 configuration register	0x00000000
R32_DMA_CNTR6	0x40020070	DMA channel 6 transfer data number register	0x00000000
R32_DMA_PADDR6	0x40020074	DMA channel 6 peripheral address register	0x00000000
R32_DMA_MADDR6	0x40020078	DMA channel 6 memory address register	0x00000000
R32_DMA_CFGR7	0x40020080	DMA channel 7 configuration register	0x00000000
R32_DMA_CNTR7	0x40020084	DMA channel 7 transfer data number register	0x00000000
R32_DMA_PADDR7	0x40020088	DMA channel 7 peripheral address register	0x00000000
R32_DMA_MADDR7	0x4002008C	DMA channel 7 memory address register	0x00000000

8.3.1 DMA Interrupt Status Register (DMA_INTFR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
27/23/19/15/11/7/3	TEIFx	RO	Transmission error flag for channel x (x=1/2/3/4/5/6/7). 1: A transmission error occurred on channel x; 0: A transmission error occurred on channel x The hardware is set, and the software writes the CTEIFx bit to clear this flag.	0
26/22/18/14/10/6/2	HTIFx	RO	Transmission more than half flag of channel x (x=1/2/3/4/5/6/7). 1: More than half of the transmission events are generated on channel x; 0: There is no more than half of the transmission on channel x. The hardware is set, and the software writes the CHTIFx bit to clear this flag.	0
25/21/17/13/9/5/1	TCIFx	RO	Transmission completion flag for channel x (x=1/2/3/4/5/6/7). 1: A transmission completion event was generated on channel x; 0: There is no transmission completion event on channel x. The hardware is set, and the software writes the CTCIFx bit to clear this flag.	0
24/20/16/12/8/4/0	GIFx	RO	Global interrupt flag for channel x（x=1/2/3/4/5/6/7）。 1: TEIFx or HTIFx or TCIFx is generated on channel x 0: No TEIFx or HTIFx or TCIFx occurs on channel x. The hardware is set, and the software writes the CGIFx bit to clear this flag.	0

8.3.2 DMA Interrupt Flag Clear Register (DMA_INTFCR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF	CHTIF	CTCIF	CGIF	CTEIF	CHTIF	CTCIF	CGIF	CTEIF	CHTIF	CTCIF	CGIF	CTEIF	CHTIF	CTCIF	CGIF

4	4	4	4	3	3	3	3	2	2	2	2	1	1	1	1

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
27/23/19/15/11/7/3	CTEIFx	WO	Clear the transmission error flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the TEIFx flag from the DMA_INTFR register; 0: No effect.	0
26/22/18/14/10/6/2	CHTIFx	WO	Clear the transmission more than half flag of channel x (x=1/2/3/4/5/6/7). 1: Clear the HTIFx flag from the DMA_INTFR register; 0: No effect.	0
25/21/17/13/9/5/1	CTCIFx	WO	Clear the transmission completion flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the TCIFx flag from the DMA_INTFR register 0: No effect.	0
24/20/16/12/8/4/0	CGIFx	WO	Clear the global interrupt flag for channel x (x=1/2/3/4/5/6/7). 1: Clear the TEIFx/HTIFx/TCIFx/GIFx flag from the DMA_INTFR register 0: No effect.	0

8.3.3 DMA Channel x Configuration Register (DMA_CFGRx) (x=1/2/3/4/5/6/7)

Offset address: 0x08 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	MEM2MEM	RW	Memory to memory mode enable. 1: Enable memory to memory data transfer mode 0: Non-memory-to-memory data transfer.	0
[13:12]	PL[1:0]	RW	Channel priority setting. 00: Low; 01: Medium. 10: High; 11: Very high.	0
[11:10]	MSIZE[1:0]	RW	Memory address data width setting. 00: 8-bit; 01: 16-bit; 10: 32-bit; 11: Reserved.	0
[9:8]	PSIZE[1:0]	RW	Peripheral address data width setting.	0

			00: 8-bit; 01: 16-bit; 10: 32-bit; 11: Reserved.	
7	MINC	RW	Memory address increment mode enable. 1: Enable memory address increment operation 0: The memory address remains unchanged.	0
6	PINC	RW	The peripheral address increment mode enables. 1: Enable peripheral address increment operation 0: The peripheral address remains unchanged.	0
5	CIRC	RW	DMA channel loop mode enable. 1: Enable cycle operation; 0: Perform a single operation.	0
4	DIR	RW	The direction of data transmission. 1: Read from memory 0: Read from the peripheral.	0
3	TEIE	RW	Transmission error interrupt enable control. 1: Enable transmission error interrupt 0: Disable transmission error interrupt.	0
2	HTIE	RW	Transmission over half interrupt enable control. 1: Enable the transmission over half interrupt. 0: Disable the transmission over half interrupt.	0
1	TCIE	RW	Transmission completion interrupt enable control. 1: Enable the transmission completion interrupt. 0: Disable the transmission completion interrupt.	0
0	EN	RW	Channel enable control. 1: Channel on; 0: Channel off. When a DMA transfer error occurs, the hardware automatically clears this bit to 0 and shuts down the channel.	0

8.3.4 DMA Channel x Number of Data Register (DMA_CNTRx) (x=1/2/3/4/5/6/7)

Offset address: 0x0C + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NDT[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	NDT[15:0]	RW	Number of data transfers, range 0-65535. Indicates the remaining number of transfers (the register contents are decremented after each DMA transfer). When the channel is in loop mode, the contents of the register are automatically reloaded to the previously	0

			configured value.	

Note: Indicates the current number of transmissions to be transmitted. When the register content is 0, no data transfer occurs regardless of whether the channel is open or not.

8.3.5 DMA Channel x Peripheral Address Register (DMA_PADDRx) (x=1/2/3/4/5/6/7)

Offset address: 0x10 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PA[31:0]																															

Bit	Name	Access	Description	Reset value
[31:0]	PA[31:0]	RW	The base address of the peripheral, which is used as the source or destination address of the peripheral data transmission. When PSIZE [1:0] = '01' (16 bits), the module automatically ignores bit0, and the operation address automatically aligns with 2 bytes; when PSIZE [1:0] ='10'(32 bits), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	0

8.3.6 DMA Channel x Memory Address Register (DMA_MADDRx) (x=1/2/3/4/5/6/7)

Offset address: 0x14 + (x-1)*20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MA[31:0]																															

Bit	Name	Access	Description	Reset value
[31:0]	MA[31:0]	RW	Memory data address, as the source or destination address for data transmission. When MSIZE [1:0] = '01' (16 bits), the module automatically ignores bit0, and the operation address automatically aligns with 2 bytes; when MSIZE [1:0] ='10'(32 bits), the module automatically ignores bit [1:0], and the operation address automatically aligns with 4 bytes.	0

Chapter 9 Analog-to-digital Converter (ADC)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The ADC module contains a 12-bit analog-to-digital converter that provides up to 8 external channels and 2 internal channels for sampling at rates up to 3Msps. It can complete the functions of single conversion, continuous conversion, automatic scanning mode, discontinuous mode, external trigger mode and so on. The channel voltage can be monitored by simulating the watchdog function. When the monitoring voltage exceeds the set threshold, the reset and protection system can be configured.

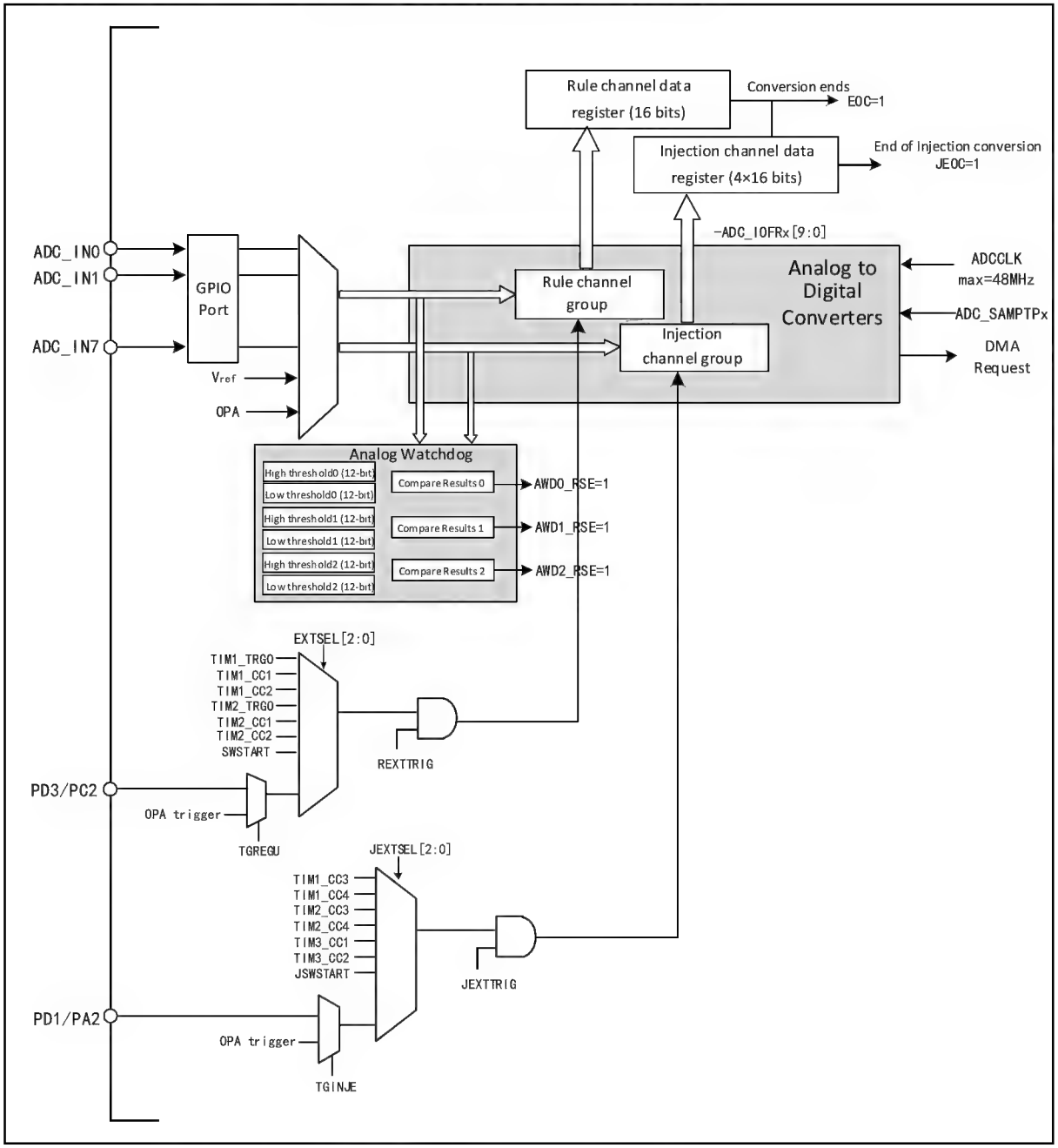
9.1 Main Features

- 12-bit resolution
- Supports 8 external channels and 2 internal signal sources for sampling
- Multiple sampling conversion methods for multiple channels: Single, continuous, scan, trigger, intermittent, etc.
- Data alignment modes: Left-aligned, right-aligned
- Sampling time can be programmed separately by channel
- Both rule conversion and injection conversion support external triggering
- Analog watchdog to monitor channel voltage
- ADC channel input range: $0 \leq V_{IN} \leq V_{DD}$

9.2 Functional Description

9.2.1 Module Structure

Figure 9-1 ADC module block diagram



9.2.2 ADC Configuration

1) Module power-up

An ADON bit of 1 in the ADC_CTLR2 register indicates that the ADC module is powered up. When the ADC module enters the power-up state (ADON=1) from the power-off mode (ADON=0), it needs to be delayed for a period of time for t_{STAB} to stabilize the module. After that, the ADON bit is written again as 1, which is used as the startup signal for the software to start the ADC conversion. By clearing the ADON bit to 0, you can terminate the current conversion and put the ADC module in power-down mode, where ADC consumes almost no power.

2) Sampling clock

The register operation of the module is based on the HBCLK (HB bus) clock, and the clock reference of its conversion unit, ADCCLK, is configured by the ADCPRE field of the RCC_CFGR0 register to divide the frequency.

3) Channel configuration

The ADC module provides 10 channel sampling sources, including 8 external channels and 2 internal channels. They can be configured into two translation groups: rule groups and injection groups. To realize the group conversion consisting of a series of transformations in any order on any number of channels.

Conversion group:

- Rule group: consists of up to 16 conversions. The rule channels and their conversion order are set in the ADC_RSQRx register. The total number of conversions in the rule group should be written to L[3:0] in the ADC_RSQR1 register.
- Injection group: consists of up to 4 conversions. The injection channels and the order of their conversions are set in the ADC_ISQR register. The total number of conversions in the injection group should be written in JL[1:0] of the ADC_ISQR register.

Note: If the ADC_RSQRx or ADC_ISQR registers are changed during conversion, the current conversion is terminated and a new start signal is sent to the ADC to convert the newly selected group.

2 internal channels.

- Vref internal reference voltage: Connected to ADC_IN8 channel.
- OPA internal output channel: Connected the ADC_IN9 channel, which is used to convert the output of OPA to numeric values.

4) Programmable sampling time

The ADC uses several ADCCLK cycles to sample the input voltage. The number of sampling cycles for a channel can be changed using the SMPx[2:0] bits in the ADC_SAMPTR1 and ADC_SAMPTR2 registers. Each channel can be sampled separately using a different time.

The total conversion time is calculated as follows.

$$T_{CONV} = \text{sampling time} + 12.5\ T_{ADCCLK}$$

The ADC's rule channel conversion supports the DMA function. The value of the rule channel conversion is stored in a data-only register, ADC_RDATAR. To prevent the data in ADC_RDATAR register from being fetched in time when multiple rule channels are converted in succession, the DMA function of ADC can be enabled. The hardware will generate a DMA request at the end of the conversion of a rule channel (EOC set) and transfer the converted data from the ADC_RDATAR register to the user-specified destination address.

After the channel configuration of the DMA controller module is completed, write DMA position 1 of the ADC_CTLR2 register to enable the DMA function of the ADC.

Note: Injection group conversion does not support DMA function.

5) Data alignment

The ALIGN bit in the ADC_CTLR2 register selects the alignment of the ADC converted data storage. 12-bit data supports left-aligned and right-aligned modes.

The data register ADC_RDATAR of the rule group channel holds the actual converted 12-bit digital value; while the data register ADC_IDATARx of the injection group channel is the actual converted data minus the value written after the offset defined in the ADC_IOFRx register, there will be positive and negative cases, so there are sign bits

(SIGNB).

Table 9-1 Data left alignment

Rule group data register

D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0	0

Injection group data register

SIGNB	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	0	0	0

Table 9-2 Data right alignment

Rule group data register

0	0	0	0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

Injection group data register

SIGNB	SIGNB	SIGNB	SIGNB	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

9.2.3 External Trigger Source

The ADC conversion start event can be triggered by an external event. If the EXTTRIG or JEXTTRIG bits of the ADC_CTLR2 register are set, the conversion of a rule group or injection group channel can be triggered by an external event, respectively. In this case, the configuration of EXTSEL[2:0] and JEXTSEL[2:0] bits determines the external event source for the rule group and injection group.

Note: When an external trigger signal is selected for ADC rule or injection conversion, only its rising edge can start the conversion.

Table 9-3 External trigger sources for rule group channels

EXTSEL[2:0]	Trigger source	Type
000	TRGO event of timer 1	Internal signal from on-chip timer
001	CC1 event of timer 1	
010	CC2 event of timer 1	
011	TRGO event of timer 2	
100	CC1 event of timer 2	
101	CC2 event of timer 2	
110	OPA trigger/ (PD3/PC2)	From external pins
111	SWSTART software trigger	Software control bits

Table 9-4 External trigger sources for injection group channels

JEXTSEL[2:0]	Trigger source	Type
000	CC3 event of timer 1	Internal signal from on-chip timer
001	CC4 event of timer 1	
010	CC3 event of timer 2	
011	CC4 event of timer 2	
100	CC1 event of timer 3	
101	CC2 event of timer 3	

110	OPA trigger/ (PD1/PA2)	From external pins
111	JSWSTART software trigger	Software control bits

9.2.4 Conversion Mode

Table 9-5 Conversion mode combinations

ADC_CTLR1 and ADC_CTLR2 register control bit					ADC conversion mode
CONT	SCAN	DISCEN/JDISCEN	JAUTO	Startup event	
0	0	0	0	ADON position 1	Single single-channel mode: A rule channel performs a single conversion.
				External trigger method	Single single-channel mode: A single conversion is performed on one of the rule channels or injection channels.
	1	0	0	ADON position 1 or external trigger method	Single scan mode: performs a single conversion of all selected rule group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence. Trigger injection method: When the rule group channel conversion process can be inserted into the injection group channel all conversion, and then continue the rule group channel conversion afterwards; but the rule group channel conversion will not be inserted when converting the injection group channel.
			1	ADON position 1 or external trigger method	Single scan mode: performs a single conversion of all selected rule group channels (ADC_RSQRx) or all injection group channels (ADC_ISQR) one by one in sequence. Automatic injection method: After the rule group channel is converted, the injection group channel is automatically converted. *Note: External trigger signals injected into the channel are not allowed during conversion.*
	0	1 (DISCEN and JDISCEN cannot both be 1)	0	External trigger method	Single intermittent mode: Each time an event is started, a short sequence (DISCNUM[2:0] defined number) of channel number transitions is executed and cannot be restarted until all selected channel transitions are completed. *Note: The DISCEN and JDISCEN control bits are selected for the rule group and injection group respectively, and the intermittent mode cannot be configured for the rule group and injection group at the same time.*

			1	-	Disable this mode.
	1	1	X	-	No such mode.
1	0	0	0	ADON position 1 or external trigger method	Continuous single channel/scan mode: repeat a new round of transitions at the end of each round until CONT clears 0 to terminate.
	1	0	0		
			1		

Note: The external trigger events for rule groups and injection groups are different, and the 'ACON' bit can only initiate Rule Group channel transitions, so the initiation events for Rule Group and Injection Group channel transitions are independent.

1) Single single-channel conversion mode

In this mode, only one conversion is performed for the current 1 channel. This mode performs conversion on the channels sorted first in the rule group or injection group, where it can be started by setting ADON position 1 of the ADC_CTLR2 register (for regular channels only) or by external triggering (for regular channels or injection channels). Once the conversion of the selected channel is completed:

If the rule group channel is converted, the conversion data is stored in the 16-bit ADC_RDATAR register, the EOC flag is set, and if the EOCIE bit is set, the ADC interrupt will be triggered.

If the injection group channel is converted, the conversion data is stored in the 16-bit ADC_IDATAR1 register, the EOC and JEOC flags are set, and if the JEOCIE or EOCIE bit is set, the ADC interrupt will be triggered.

2) Single scan mode conversion

Enter ADC scan mode by setting the scan bit of the ADC_CTLR1 register to 1. This mode is used to scan a set of analog channels and perform a single conversion one by one for all channels selected by the ADC_RSQRx register (for regular channels) or ADC_ISQR (for injection channels). When the current channel conversion ends, the next channel of the same group is automatically converted.

In the scanning mode, according to the state of JAUTO bits, it can be divided into trigger injection mode and automatic injection mode.

- Trigger injection

The JAUTO bit is 0, when the trigger event of the injection group channel conversion occurs in the process of scanning the rule group channel, the current conversion is reset, and the sequence of the injection channel is carried out in a single scan mode. After all the selected injection group channel scan conversion is completed, the last interrupted rule group channel conversion is restored.

If the start event of the regular channel occurs when scanning the channel sequence of the injection group, the conversion of the injection group will not be interrupted, but the conversion of the rule sequence will be performed after the conversion of the injection sequence is completed.

Note: When using triggered injection transformations, you must ensure that the interval between triggered events is longer than the injection sequence. For example, if it takes 28 ADCCLK to complete the conversion of the injection sequence, the minimum time between events to trigger the injection channel is 29 ADCCLK.

- Auto-injection

The JAUTO bit is 1, and after scanning all the channel translations selected by the rule group, the channel selected by the injection group is converted automatically. This method can be used to convert up to 20 conversion sequences in ADC_RSQRx and ADC_ISQR registers.

In this mode, the external trigger (JEXTTRIG=0) of the injection channel must be disabled.

Note: When the ADC clock pre-division factor (ADCPRE) is 4 to 8, 1 ADCCLK interval is automatically inserted

when switching from regular conversion to injection sequence or from injection conversion to regular sequence; when the ADC clock pre-division factor is 2, there is a delay of 2 ADCCLK intervals.

3) Single intermittent mode conversion

Enter the break mode of the rule group or injection group by setting the DISCEN or JDISCEN bit of the ADC_CTLR1 register to 1. This mode distinguishes scanning a complete set of channels in the scan mode, but divides a group of channels into multiple short sequences, and each external trigger event will perform a scan conversion of a short sequence.

The length of the short sequence n (n < = 8) is defined in the DISCNUM [2:0] of the ADC_CTLR1 register. When DISCEN is 1, it is the discontinuous mode of the rule group, and the total length to be converted is defined in the L [3:0] of the ADC_RSQR1 register. When JDISCEN is 1, it is the discontinuous mode of the injection group, and the total length to be converted is defined in the JL [1:0] of the ADC_ISQR register. Both rule group and injection group cannot be set to discontinuous mode at the same time.

Example of rule group discontinuity mode:

DISCEN=1, DISCNUM[2:0]=3, L[3:0]=8, channels to be converted=1, 3, 2, 5, 8, 4, 10, 6

The 1st external trigger: conversion sequence is: 1, 3, 2

The 2nd external trigger: conversion sequence is: 5, 8, 4

The 3rd external trigger: conversion sequence is: 10, 6, while generating EOC events

The 4th external trigger: conversion sequence is: 1, 3, 2

Example of injection group intermittent mode:

JDISCEN=1, DISCNUM[2:0]=1, JL[1:0]=3, channels to be converted=1, 3, 2

The 1st external trigger: conversion sequence is: 1

The 2nd external trigger: the conversion sequence is: 3

The 3rd external trigger: conversion sequence is: 2, generating both EOC and JEOC events

The 4th external trigger: conversion sequence is: 1

Note: 1. When a rule group or injection group is converted in intermittent mode, the conversion sequence does not automatically start from scratch at the end of the conversion sequence. When all subgroups are converted, the next trigger event initiates the conversion of the first subgroup.

2. You cannot use auto-injection (JAUTO=1) and intermittent mode at the same time.

3. You cannot set intermittent mode for both rule groups and injection groups, and intermittent mode can only be used for a group of conversions.

4) Continuous conversion

In the continuous conversion mode, another conversion is started as soon as the previous ADC conversion is completed. The conversion will not stop on the last channel of the selection group, but will continue conversion from the first channel of the selection group again. The startup events in this mode include external trigger events and the ADON bit is set to 1. After setting the startup, the CONT bit needs to be set to 1.

If a rule channel is converted, the conversion data is stored in the ADC_RDATAR register, the conversion end flag EOC is set, and if EOCIE is set, an interrupt is generated.

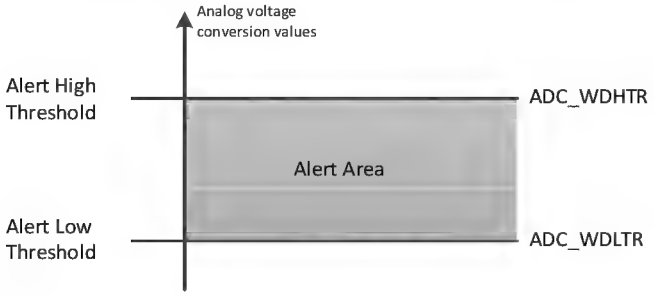
If an injection channel is converted, the conversion data is stored in the ADC_IDATARx register, the injection conversion end flag JEOC is set, and if JEOCIE is set, an interrupt is generated.

9.2.5 Analog Watchdog

If the analog voltage converted by the ADC is lower than the low threshold or higher than the high threshold, the AWD analog watchdog status bit is set. The threshold setting is located in the lowest 12 significant bits of the

ADC_WDHTR and ADC_WDLTR registers. The corresponding interrupt is allowed by setting the AWDIE bit of the ADC_CTLR1 register.

Figure 9-2 Analog watchdog threshold area



Configure the AWDSGL, AWDEN, JAWDEN and AWDCH[4:0] bits of the ADC_CTLR1 register to select the channel for analog watchdog alerting, as related in the following table.

Table 9-6 Analog Watchdog channel selection

Analog watchdog alert channel	ADC_CTLR1 register control bit			
	AWDSGL	AWDEN	JAWDEN	AWDCH[4:0]
No vigilance	Ignore	0	0	Ignore
All injection channels	0	0	1	Ignore
All rule channels	0	1	0	Ignore
All injection and rule channels	0	1	1	Ignore
Single injection channel	1	0	1	Determine the channel number
Single rule channel	1	1	0	Determine the channel number
Single injection and rule channel	1	1	1	Determine the channel number

9.3 Register Description

Table 9-7 ADC-related registers list

Name	Access address	Description	Reset value
R32_ADC_STATR	0x40012400	ADC Status Register	0x00000000
R32_ADC_CTLR1	0x40012404	ADC Control Register 1	0x00000000
R32_ADC_CTLR2	0x40012408	ADC Control Register 2	0x00000000
R32_ADC_SAMPTR2	0x40012410	ADC Sample Time Configuration Register 2	0x00000000
R32_ADC_IOFR1	0x40012414	ADC Injection Channel Data Offset Register 1	0x00000000
R32_ADC_IOFR2	0x40012418	ADC Injection Channel Data Offset Register 2	0x00000000

R32_ADC_IOFR3	0x4001241C	ADC Injection Channel Data Offset Register 3	0x00000000
R32_ADC_IOFR4	0x40012420	ADC Injection Channel Data Offset Register 4	0x00000000
R32_ADC_WDHTR	0x40012424	ADC Watchdog High Threshold Register	0x00000FFF
R32_ADC_WDLTR	0x40012428	ADC Watchdog Low Threshold Register	0x00000000
R32_ADC_RSQR1	0x4001242C	ADC Rule Sequence Register 1	0x00000000
R32_ADC_RSQR2	0x40012430	ADC Rule Sequence Register 2	0x00000000
R32_ADC_RSQR3	0x40012434	ADC Rule Sequence Register 3	0x00000000
R32_ADC_ISQR	0x40012438	ADC Injection Sequence Register	0x00000000
R32_ADC_IDATAR1	0x4001243C	ADC Injection Data Register 1	0x00000000
R32_ADC_IDATAR2	0x40012440	ADC Injection Data Register 2	0x00000000
R32_ADC_IDATAR3	0x40012444	ADC Injection Data Register 3	0x00000000
R32_ADC_IDATAR4	0x40012448	ADC Injection Data Register 4	0x00000000
R32_ADC_RDATAR	0x4001244C	ADC Rule Data Register	0x00000000
R32_ADC_CTLR3	0x40012450	ADC Control Register 3	0x00000001
R32_ADC_WDTR1	0x40012454	ADC Watchdog 1 Threshold Register	0x0FFF0000
R32_ADC_WDTR2	0x40012458	ADC Watchdog 2 Threshold Register	0x0FFF0000

9.3.1 ADC Status Register (ADC_STATR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved											STRT	JSTRT	JEOC	EOC	AWD

Bit	Name	Access	Description	Reset value
[31:5]	Reserved	RO	Reserved	0
4	STRT	RW0	Rule channel transition start state. 1: Rule channel conversion has started. 0: Rule channel conversion is not started. This bit is set to 1 by hardware and cleared to 0 by software (write 1 is not valid).	0
3	JSTRT	RW0	Injection channel conversion start state. 1: Injection channel conversion has started. 0: Injection channel conversion has not started. This bit is set to 1 by hardware and cleared to 0 by software (write 1 is not valid).	0
2	JEOC	RW0	Injection channel group conversion completed status. 1: Conversion completed; 0: Conversion is not completed. This bit is set to 1 by hardware (all injected channels are converted) and cleared to 0 by software (write 1 is invalid).	0
1	EOC	RW0	Conversion end status.	0

			1: Conversion completed; 0: Conversion is not completed. This bit is set to 1 by hardware (end of rule or injection channel group conversion), cleared by software to 0 (write 1 is invalid) or when reading ADC_RDATAR.	
0	AWD	RW0	Analog watchdog flag bit. 1: Occurrence of analog watchdog events. 0: No analog watchdog event occurred. This bit is set to 1 by hardware (conversion value is out of range of ADC_WDHTR and ADC_WDLTR registers) and cleared to 0 by software (write 1 is not valid).	0

9.3.2 ADC Control Register 1 (ADC_CTLR1)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved					BUFEN	TKITUNE	TKENABLE	AWDEN	JAWDEN	Reserved					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DISCNUM[2:0]			JDISCEN	DISCEN	JAUTO	AWDSGL	SCAN	JEOCIE	AWDIE	EOCIE	AWDCH[4:0]				

Bit	Name	Access	Description	Reset value
[31:27]	Reserved	RO	Reserved	0
26	BUFEN	RW	ADC BUF enable. 1: Enable; 0: Disable.	0
25	TKITUNE	RW	Touchkey current adjustment. 1: 17.5μA; 0: 35μA.	0
24	TKENABLE	RW	TKEY module enable control. 1: Enable TKEY module; 0: Disable TKEY module.	0
23	AWDEN	RW	Analog watchdog function enable bit on the rule channel. 1: Enable analog watchdog on the rule channel; 0: Disable analog watchdog on the rule channel.	0
22	JAWDEN	RW	Analog watchdog function enable bit on the injection channel. 1: Enable analog watchdog on the injection channel; 0: Disable analog watchdog on the injection channel.	0
[21:16]	Reserved	RO	Reserved	0
[15:13]	DISCNUM[2:0]	RW	Number of rule channels to be converted after external triggering in intermittent mode. 000: 1 channel.	0

			... 111: 8 channels.	
12	JDISCEN	RW	Intermittent mode enable bit on the injection channel. 1: Enable intermittent mode on the injection channel; 0: Disable intermittent mode on the injection channel.	0
11	DISCEN	RW	Intermittent mode enable bit on the rule channel. 1: Enable intermittent mode on the rule channel; 0: Disable intermittent mode on the rule channel.	0
10	JAUTO	RW	Upon completion of opening a rule channel, the injection channel group enable bit is automatically converted. 1: Enable automatic injection channel group conversion; 0: Disable automatic injection channel group conversion. *Note: This mode requires disabling the external trigger function of the injection channel.*	0
9	AWDSGL	RW	In scan mode, use the analog watchdog enable bit on a single channel. 1: Use analog watchdog on a single channel (AWDCH[4:0] selection); 0: Use analog watchdog on all channels.	0
8	SCAN	RW	Scan mode enable bit. 1: Enable scan mode (Continuously convert all channels selected by ADC_IOFRx and ADC_RSQRx); 0: Disable scan mode.	0
7	JEOCIE	RW	The injection channel group conversion ends the interrupt enable bit. 1: Enable injection channel group conversion completion interrupt (JEOC flag); 0: Disable injection channel group conversion completion interrupt.	0
6	AWDIE	RW	Analog watchdog interrupt enable bit. 1: Enable analog watchdog interrupt; 0: Disable analog watchdog interrupt. *Note: In scan mode, if this interruption occurs, the scan will be aborted.*	0
5	EOCIE	RW	The conversion ends (rule or injection channel group) interrupt enable bit. 1: Enable conversion end interrupt (EOC flag) 0 Disable conversion end interrupt.	0
[4:0]	AWDCH[4:0]	RW	Analog watchdog channel selection bit. 00000: Analog input channel 0; 00001: Analog input channel 1; … 01001: Analog input channel 9.	0

9.3.3 ADC Control Register 2 (ADC_CTLR2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved									SWSTART	JSWSTART	EXTTRIG	EXTSEL[2:0]			Reserved

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JEXTTRIG	JEXTSEL[2:0]			ALIGN	Reserved		DMA	Reserved		TGINJE	TGREGU	Reserved		CONT	ADON

Bit	Name	Access	Description	Reset value
[31:23]	Reserved	RO	Reserved	0
22	SWSTART	RW	Initiates a rule channel conversion, need to set software trigger. 1: Startup rule channel conversion; 0: Reset status. This bit is set by software and cleared to 0 by hardware when conversion starts.	0
21	JSWSTART	RW	Initiates an injection channel conversion, need to set software trigger. 1: Startup injection channel conversion; 0: Reset status. This bit is set by software and cleared to 0 by hardware when conversion starts.	0
20	EXTTRIG	RW	External trigger conversion mode enabled on rule channel. 1: Enable the external event startup function; 0: Disable the external event startup function.	0
[19:17]	EXTSEL[2:0]	RW	Initiates external trigger event selection of rule channel conversion. 000: TRGO event of timer 1; 001: CC1 event of timer 1; 010: CC2 event of timer 1; 011: TRGO event of timer 2; 100: CC1 event of timer 2; 101: CC2 event of timer 2; 110: OPA trigger/ (PD3/PC2); 111: SWSTART software trigger.	0
16	Reserved	RO	Reserved	0
15	JEXTTRIG	RW	External trigger conversion mode enabled of injection channel. 1: Use external events to initiate conversions; 0: Disable external event initiation.	0
[14:12]	JEXTSEL[2:0]	RW	Initiates external trigger event selection of injection channel conversion.	0

			000: CC3 event of timer 1; 001: CC4 event of timer 1; 010: CC3 event of timer 2; 011: CC4 event of timer 2; 100: CC1 event of timer 3; 101: CC2 event of timer 3; 110: OPA trigger/ (PD1/PA2); 111: JSWSTART software trigger.	
11	ALIGN	RW	Data alignment. 1: Left-aligned; 0: Right-aligned.	0
[10:9]	Reserved	RO	Reserved	0
8	DMA	RW	Direct Memory Access (DMA) mode enable. 1: Enable DMA mode. 0: Disable DMA mode.	0
[7:6]	Reserved	RO	Reserved	0
5	TGINJE	RW	External trigger event of injection channel conversion (JEXTSEL[2:0]=110) OPA trigger selection bit. 1: OPA trigger; 0: PD1/PA2.	0
4	TGREGU	RW	External trigger event of rule channel conversion (EXTSEL[2:0]=110) OPA trigger selection bit. 1: OPA trigger; 0: PD3/PC2.	0
[3:2]	Reserved	RO	Reserved	0
1	CONT	RW	Continuous conversion enable. 1: Continuous conversion mode; 0: Single conversion mode. If this bit is set, conversion will be continuous until the bit is cleared.	0
0	ADON	RW	Turns the A/D converter on/off. When this bit is 0, writing 1 will wake up the ADC from power-down mode; when this bit is 1, writing 1 will start the conversion. 1: Turns the ADC on and initiates conversion; 0: turns off ADC conversion/calibration and enters power-down mode. *Note: A conversion is initiated when only ADON is changed in the register; if there are any other arbitrary bits sent for change, a new conversion will not be initiated.*	0

9.3.4 ADC Sample Time Configuration Register 2 (ADC_SAMPTR2)

Offset address: 0x10

31 30	29 28 27	26 25 24	23 22 21	20 19 18	17 16
Reserved	SMP9[2:0]	SMP8[2:0]	SMP7[2:0]	SMP6[2:0]	SMP5[2:1]

15	14 13 12	11 10 9	8 7 6	5 4 3	2 1 0
SMP5[0]	SMP4[2:0]	SMP3[2:0]	SMP2[2:0]	SMP1[2:0]	SMP0[2:0]

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:0]	SMPx[2:0]	RW	SMPx[2:0] (x=0-9) Sampling time configuration of channel x. In low power mode: 000: 3.5 cycles; 001: 7.5 cycles; 010: 13.5 cycles; 011: 28.5 cycles; 100: 41.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; In non-low power mode: 000: 3.5 cycles; 001: 7.5 cycles; 010: 11.5 cycles; 011: 19.5 cycles; 100: 35.5 cycles; 101: 55.5 cycles; 110: 71.5 cycles; 111: 239.5 cycles; These bits are used to select the sampling time of each channel independently, and the channel configuration value must remain unchanged during the sampling period.	0

9.3.5 ADC Injection Channel Data Offset Register x (ADC_IOFRx) (x=1/2/3/4)

Offset address: 0x14 + (x-1)*4

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved

15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Reserved	JOFFSETx[11:0]

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	JOFFSETx[11:0]	RW	The data offset value of the injection channel x. When converting the injection channel, this value defines the value to be subtracted from the original conversion data. The result of the conversion can be read out in the ADC_IDATARx register.	0

9.3.6 ADC Watchdog High Threshold Register (ADC_WDHTR)

Offset address: 0x24

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
Reserved

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	HT[11:0]

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	HT[11:0]	RW	Analog watchdog high threshold setting value.	FFFh

Note: You can change the values of WDHTR and WDLTR during the conversion process, but they will take effect the next time you convert.

9.3.7 ADC Watchdog Low Threshold Register (ADC_WDLTR)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LT[11:0]											

Bit	Name	Access	Description	Reset value
[31:12]	Reserved	RO	Reserved	0
[11:0]	LT[11:0]	RW	Analog watchdog low threshold setting value.	0

Note: You can change the values of WDHTR and WDLTR during the conversion process, but they will take effect the next time you convert.

9.3.8 ADC Rule Sequence Register 1 (ADC_RSQR1)

Offset address: 0x2C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								L[3:0]				SQ16[4:1]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ16[0]	SQ15[4:0]					SQ14[4:0]					SQ13[4:0]				

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:20]	L[3:0]	RW	The number of channels that need to be converted in the regular channel conversion sequence. 0000-1111: 1-16 conversions.	0
[19:15]	SQ16[4:0]	RW	The number of the 16th conversion channel in the rule sequence (0-9).	0
[14:10]	SQ15[4:0]	RW	The number of the 15th conversion channel in the rule sequence (0-9).	0
[9:5]	SQ14[4:0]	RW	The number of the 14th conversion channel in the rule sequence (0-9).	0
[4:0]	SQ13[4:0]	RW	The number of the 13th conversion channel in the rule sequence (0-9).	0

9.3.9 ADC Rule Sequence Register 2 (ADC_RSQR2)

Offset address: 0x30

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ12[4:0]					SQ11[4:0]					SQ10[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ10[0]		SQ9[4:0]					SQ8[4:0]					SQ7[4:0]			

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ12[4:0]	RW	The number of the 12th conversion channel in the rule sequence (0-9).	0
[24:20]	SQ11[4:0]	RW	The number of the 11th conversion channel in the rule sequence (0-9).	0
[19:15]	SQ10[4:0]	RW	The number of the 10th conversion channel in the rule sequence (0-9).	0
[14:10]	SQ9[4:0]	RW	The number of the 9th conversion channel in the rule sequence (0-9).	0
[9:5]	SQ8[4:0]	RW	The number of the 8th conversion channel in the rule sequence (0-9).	0
[4:0]	SQ7[4:0]	RW	The number of the 7th conversion channel in the rule sequence (0-9).	0

9.3.10 ADC Rule Sequence Register 3(ADC_RSQR3)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		SQ6[4:0]					SQ5[4:0]					SQ4[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4[0]		SQ3[4:0]					SQ2[4:0]					SQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:30]	Reserved	RO	Reserved	0
[29:25]	SQ6[4:0]	RW	The number of the 6th conversion channel in the rule sequence (0-9).	0
[24:20]	SQ5[4:0]	RW	The number of the 5th conversion channel in the rule sequence (0-9).	0
[19:15]	SQ4[4:0]	RW	The number of the 4th conversion channel in the rule sequence (0-9).	0
[14:10]	SQ3[4:0]	RW	The number of the 3rd conversion channel in the rule sequence (0-9).	0
[9:5]	SQ2[4:0]	RW	The number of the 2nd conversion channel in the rule sequence (0-9).	0

[4:0]	SQ1[4:0]	RW	The number of the 1st conversion channel in the rule sequence (0-9).	0

9.3.11 ADC Injection Sequence Register (ADC_ISQR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved										JL[1:0]		JSQ4[4:1]			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ4[0]	JSQ3[4:0]					JSQ2[4:0]						JSQ1[4:0]			

Bit	Name	Access	Description	Reset value
[31:22]	Reserved	RO	Reserved	0
[21:20]	JL[1:0]	RW	The number of channels that need to be converted in the injection channel conversion sequence. 00-11: 1-4 conversions.	0
[19:15]	JSQ4[4:0]	RW	The number of the 4th conversion channel in the injection sequence (0-9). *Note: the software writes and assigns the channel number (0-10) to the 4th of the sequence to be converted.*	0
[14:10]	JSQ3[4:0]	RW	The number of the 3rd conversion channel in the injection sequence (0-9).	0
[9:5]	JSQ2[4:0]	RW	The number of the 2nd conversion channel in the injection sequence (0-9).	0
[4:0]	JSQ1[4:0]	RW	The number of the 1st conversion channel in the injection sequence (0-9).	0

Note: Unlike the regular conversion sequence, if the length of JL[1:0] is less than 4, the sequence order of conversion starts from (4 - JL).

For example, when JL[1:0]=3 (4 injected transitions in the sequencer), the ADC will convert channels in the following order: JSQ1[4:0], JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0];

When JL[1:0]=2 (3 injected transitions in the sequencer), the ADC will convert the channels in the following order: JSQ2[4:0], JSQ3[4:0] and JSQ4[4:0];

When JL[1:0]=1 (2 injected conversions in the sequencer), the ADC converts the channels in the following order: first JSQ3[4:0], then JSQ4[4:0];

When JL[1:0] = 0 (1 injection conversion in the sequencer), the ADC will convert only the JSQ4[4:0] channels.

If ADCx_ISQR[21:0]=10 00111 00011 00111 00010, the ADC will convert channels in the following order: JSQ2[4:0], JSQ3[4:0], and JSQ4[4:0], indicating that the scan conversions are performed in the following channel order: 7, 3, 7.

9.3.12 ADC Injection Data Register (ADC_IDATARx) (x=1/2/3/4)

Offset address: 0x3C + (x-1)*4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JDATA[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	JDATA[15:0]	RO	Injection channel conversion data (data left- aligned or right-aligned).	0

9.3.13 ADC Rule Data Register (ADC_RDATAR)

Offset address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	DATA[15:0]	RO	Rule channel conversion data (data left- aligned or right-aligned).	0

9.3.14 ADC Control Register (ADC_CTLR3)

Offset address: 0x50

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								DRV_OUTEN							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					AWD2_RES	AWD1_RES	AWD0_RES	Reserved	AWD2_RST_EN	AWD1_RST_EN	AWD0_RST_EN	AWD_SCAN	DRVEN	DUTYEN	ADC_LP

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
[23:16]	DRV_OUTEN	RW	Touchkey multi-mask each channel enable.	0
[15:11]	Reserved	RO	Reserved	0
10	AWD2_RES	RW0	Analog watchdog 2 compare result. 1: The conversion value is greater than the high threshold of watchdog 2 or less than the low threshold of watchdog 2. 0: The conversion value is between the high threshold and the low threshold of watchdog 2. *Note: Hardware is set to 1, write 0 to clear zero.*	0
9	AWD1_RES	RW0	Analog watchdog 1 compare result. 1: The conversion value is greater than the high threshold of	0

			watchdog 1 or less than the low threshold of watchdog 1. 0: The conversion value is between the high threshold and the low threshold of watchdog 1. *Note: Hardware is set to 1, write 0 to clear zero.*	
8	AWD0_RES	RW0	Analog watchdog 0 compare result. 1: The conversion value is greater than the high threshold of watchdog 0 or less than the low threshold of watchdog 0. 0: The conversion value is between the high threshold and the low threshold of watchdog 0. *Note: Hardware is set to 1, write 0 to clear zero.*	0
7	Reserved	RO	Reserved	0
6	AWD2_RST_EN	RW	Analog watchdog 2 output reset enable bit: 1: Enable; 0: Disable.	0
5	AWD1_RST_EN	RW	Analog watchdog 1 output reset enable bit: 1: Enable; 0: Disable.	0
4	AWD0_RST_EN	RW	Analog watchdog 0 output reset enable bit: 1: Enable; 0: Disable.	0
3	AWD_SCAN	RW	Analog watchdog scan enable: 1: Enable watchdog scan; 0: Disable watchdog scan.	0
2	DRVEN	RW	Touchkey multi-mask enable: 1: Enable; 0: Disable.	0
1	DUTYEN	RW	ADC clock duty cycle control bit: 1: The duty cycle of ADC clock can be adjusted by delay, and the delay of high level can be 1ns. 0: The ADC clock is configured by RCC with a duty cycle of 50% or 75%.	0
0	ADC_LP	RW	ADC low power mode control bits: 1: Low power mode, suitable for sampling rate below 1M 0: High power consumption, suitable for 1m and above sampling rate, and only for voltage above 4.5V.	1

9.3.15 ADC Watchdog 1 Threshold Register (ADC_WDTR1)

Offset address: 0x54

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				HTR1											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LTR1											

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
[27:16]	HTR1	RW	Analog watchdog high threshold setting value.	0xFFF
[15:12]	Reserved	RO	Reserved	0
[11:0]	LTR1	RW	Analog watchdog low threshold setting value.	0x000

Note: Only applicable to watchdog channel 1.

9.3.16 ADC Watchdog 2 Threshold Register (ADC_WDTR2)

Offset address: 0x58

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				HTR2											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved				LTR2											

Bit	Name	Access	Description	Reset value
[31:28]	Reserved	RO	Reserved	0
[27:16]	HTR2	RW	Analog watchdog high threshold setting value.	0xFFF
[15:12]	Reserved	RO	Reserved	0
[11:0]	LTR2	RW	Analog watchdog low threshold setting value.	0x000

Note: Only applicable to watchdog channel 2.

Chapter 10 Touch Key Detection (TKEY)

This chapter applies to the CH32V006, CH32V007, CH32M007.

The touch detection control (TKEY) unit, with the help of the voltage conversion function of the ADC module, realizes the touch key detection function by converting the capacitance to the voltage for sampling. The detection channel reuses the 8 external channels of ADC, and the touch key detection is realized through the single conversion mode of the ADC module.

10.1 Functional Description

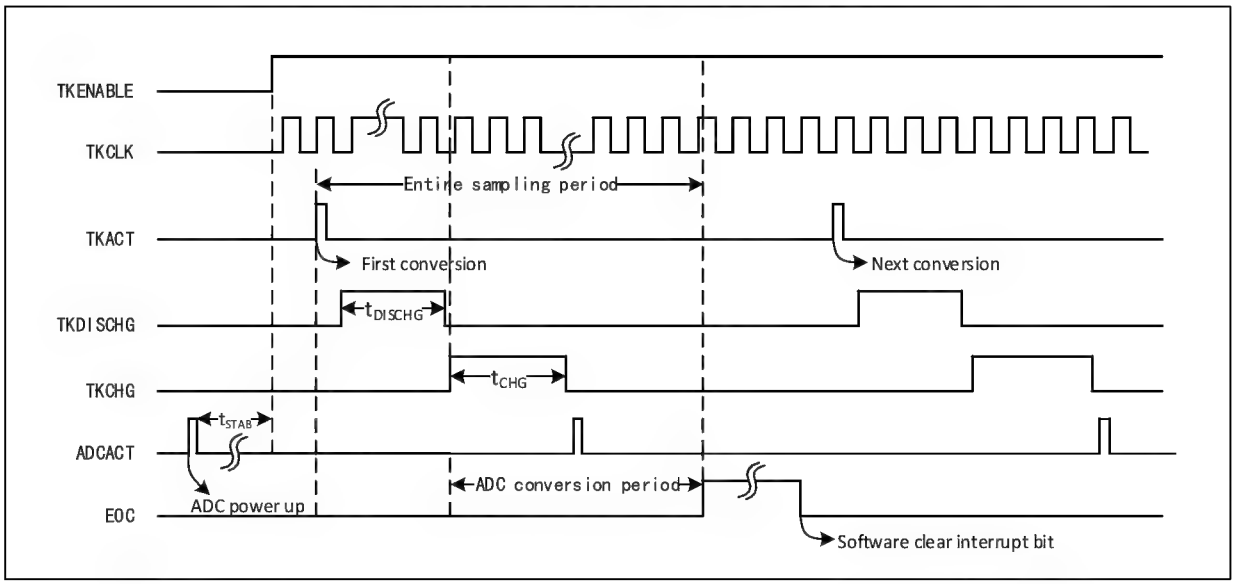
- Enable TKEY

The TKEY detection process needs the cooperation of the ADC module, so when using the TKEY function, it is necessary to ensure that the ADC module is in the power-on state (ADON=1), and then turn on the TKEY unit function by changing the TKENABLE position 1 of the ADC_CTLR1 register, and the charging current of the TKEY module can be adjusted through the TKITUNE bit.

TKEY only supports single channel conversion mode. Configure the channel to be converted to the first rule group sequence of the ADC module, and the software starts the conversion (Write R32_TKEY_DISCHG register).

Note: When the TKEY conversion is disabled, ADC channel configuration function can still be retained.

Figure 10-1 TKEY working timing diagram



- Programmable sampling time

TKEY unit conversion needs to use several HBCLK clock cycles (t_{DISCHG}) for discharge, and then charge and sample the voltage of the channel through several HBCLK cycles (t_{CHG}). The number of discharge cycles is configured by register R32_TKEY_DISCHG, the number of charging cycles is configured by register R32_TKEY_CHG, and all channels use the same number of charge and discharge cycles.

10.2 TKEY Operations

TKEY detection is an extended function of ADC module. Its working principle is to change the capacitance sensed by the hardware channel through "touch" and "non-touch" methods, and then to convert the capacitance change into

the voltage change and finally convert into a digital value by the ADC module.

During sample, ADC needs to be configured as a single 1-channel working mode, and a conversion is started by the "write operation" of the R32_TKEY_DISCHG register. The specific process is as follows:

1) Initialize the ADC function, configure the ADC module as a single conversion module, set the ACON bit to 1, and wake up the ADC module. Set the TKENABLE bit in the ADC_CTLR1 register to 1, and switch on the TKEY unit.

2) Set the channel to be converted, write the channel serial number into the first conversion position in the ADC regular group sequence (ADC_RSQR3[4:0]), and set L[3:0] to 1.

3) Set the charge sample time of the channel, write to the R32_TKEY_CHG register, unit: HBCLK.

4) Write to R32_TKEY_DISCHG, set the discharge time, unit: HBCLK, to start a TKEY sample and conversion.

5) Wait for the EOC conversion end flag bit in the ADC status register to be set to 1, read the ADC_DR register to obtain the conversion value.

6) To perform next conversion, repeat steps 2-5.

10.3 TKEY Register Description

Table 10-1 TKEY1 registers

Name	Access address	Description	Reset value
R32_TKEY_DISCHG	0x4001243C	TKEY Charge Time Configuration Register	0x00000000
R32_TKEY_CHG	0x4001244C	TKEY Start and Discharge Time Configuration Register	0x00000000
R32_TKEY_DR	0x4001244C	TKEY Data Register	0x00000000

10.3.1 TKEY Charge Time Configuration Register (R32_TKEY_CHG)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TKCHARGE[10:0]										

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
[10:0]	TKCHARGE[10:0]	WO	TKEY charge time (Unit: system clock cycle)	0

Note: This register maps the injection data register 1 (ADC_IDATAR1) of the ADC module. Therefore, when the address register performs a "write operation", it is executed as a TKEY charging time (R32_TKEY_CHG); when a "read operation" is performed, it is executed as an injection data register 1 (ADC_IDATAR1) of the ADC module.

10.3.2 TKEY Start and Discharge Time Configuration Register (R32_TKEY_DISCHG)

Offset address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					TKACT_DCG[10:0]										

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
[10:0]	TKACT_DCG[10:0]	WO	Write the discharge time and start a TKEY channel detection.	0

Note: This register maps the rule data register (ADC_RDATAR) of the ADC module.

10.3.3 TKEY Data Register (R32_TKEY_DR)

Offset address: 0x4C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:0]	DATA[15:0]	RO	Converted data.	0

Note: This register maps the rule data register (ADC_RDATAR) of the ADC module.

Chapter 11 Advanced-control Timer (ADTM)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The Advanced Timer Module contains a powerful 16-bit auto-reload timer, TIM1, which can be used to measure pulse widths or to generate pulses, PWM waves, and more. It is used in motor control, power supply and other fields. CH32M007 has a built-in gate-level driver for three-phase motor power tubes, which is controlled by an advanced timer TIM1 that generates a 6-channel PWM signal, and should be used by setting TIM1_RM=0100 to pin-map timer 1.

11.1 Main Features

The main features of the advanced timer TIM1 include:

- 16-bit automatic reinstall counter, supporting increasing counting mode, decreasing counting mode and increasing and decreasing counting mode
- 16-bit prescaler, frequency division coefficient is dynamically adjustable from 1 to 65536
- Support for four independent comparison capture channels.
- Each comparison capture channel supports multiple operating modes, such as: input capture, output comparison, PWM generation and single pulse output.
- Complementary outputs supporting programmable dead-time.
- Support external signals to control the timer.
- Support updating the timer after a defined period using a repeat counter.
- Support resetting the timer or placing it in the OK state using the brake signal.
- Support the use of DMA in multiple modes.
- Support incremental encoders.
- Support cascading and synchronization between timers.

11.2 Principle and Structure

This section deals with the internal construction of advanced-control timers.

11.2.1 Overview

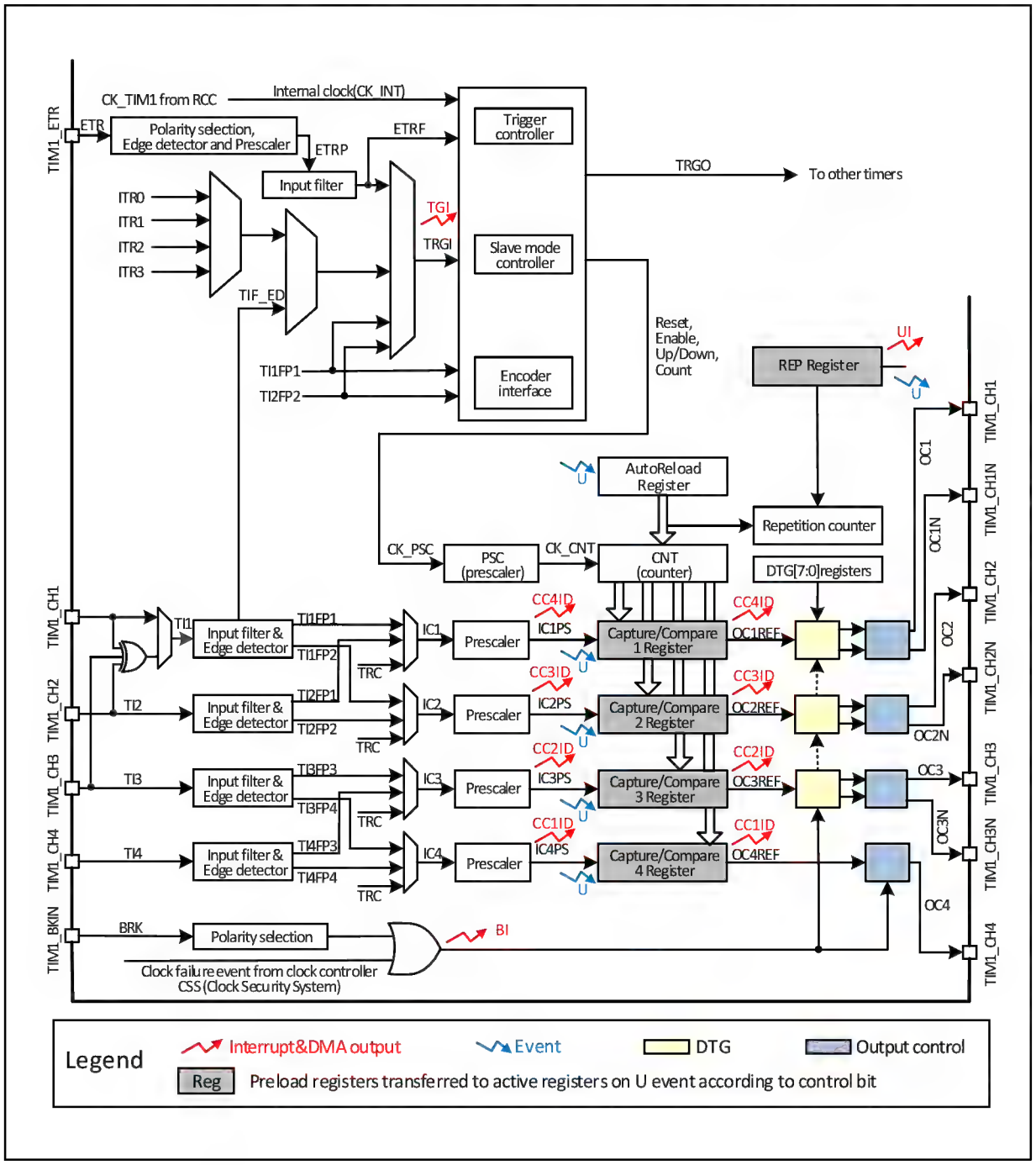
As shown in figure 11-1, the structure of the advanced timer can be roughly divided into three parts, namely, the input clock part, the core counter part and the comparison capture channel part.

The clock of the advanced-control timer can come from the HB bus clock (CK_INT), from the external clock input pin (TIM1_ETR), from other timers with clock output function (ITRx), or from the input of the comparison capture channel (TIM1_CHx). After various set filtering and frequency division operations, these input clock signals become CK_PSC clocks and output to the core counter. In addition, these complex clock sources can also be output as TRGO to other timers and ADC peripherals.

The core of the advanced timer is a 16-bit counter (CNT). After being divided by the prescaler (PSC), the CK_PSC becomes CK_CNT and output to CNT, CNT to support the increasing counting mode, decreasing counting mode and increasing or decreasing counting mode, and has an automatic reload value register (ATRLR) to reload the initial values for the CNT at the end of each counting cycle. There is also an auxiliary counter that counts the number of times ATRLR reloads initial values for CNT, which can generate specific events when the number of times reaches the number of times set in the repeated count register (RPTCR).

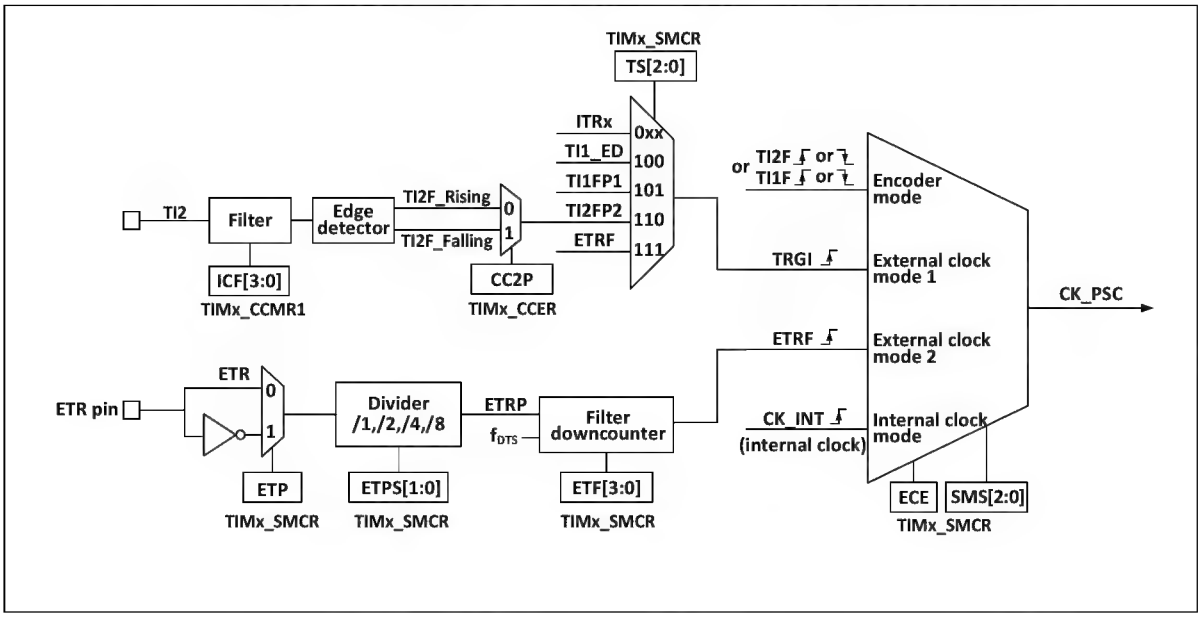
The advanced-control timer has four sets of comparison capture channels, each of which can input pulses from its own pin or output waveform to the pin, that is, the compare/capture channel supports input and output modes. The input of each channel of the compare/capture register supports operations such as filtering, frequency division and edge detection, supports mutual trigger between channels, and provides a clock for the core counter CNT. Each comparison acquisition channel has a set of comparison acquisition registers (CHxCVR) that support comparison with the master counter (CNT) to output pulses.

Figure 11-1 Block diagram of advanced-control timer structure



11.2.2 Clock Input

Figure 11-2 Block diagram of CK_PSC source for advanced-control timer



Advanced-control timer CK_PSC has many clock sources and can be divided into four categories:

1) The external clock pin (ETR) inputs the route of the clock: ETR→ETRP→ETRF;
2) Internal HB clock input route: CK_INT;
3) Route from the comparison capture channel pin (TIM1_CHx): TIM1_CHx→TIx→TIxFPx, this route is also used in encoder mode;
4) Input from other internal timers: ITRx;

By determining the input pulse selection of the SMS from which the CK_PSC comes from, the actual operation can be divided into four categories:

1) Select internal clock source (CK_INT);
2) External clock source mode 1;
3) External clock source mode 2;
4) Encoder mode;

All 4 clock sources mentioned above can be selected by these four operations.

11.2.2.1 Internal clock source (CK_INT)

If the advanced timer is started when the SMS domain is kept at 000b, then the internal clock source (CK_INT) is selected as the clock. At this point, CK_INT is CK_PSC.

11.2.2.2 External Clock Source Mode 1

If the SMS domain is set to 111b, external clock source mode 1 is enabled. When the external clock source 1 is enabled, TRGI is selected as the source of CK_PSC, and it is worth noting that you also need to configure the TS domain to select the source of TRGI. The following pulses can be selected as clock sources in the TS domain:

1) Internal trigger (ITRx, x=0, 1, 2, 3)
2) The signal after compare/capture channel 1 through the edge detector (TI1F_ED).
3) The signal TI1FP1, TI2FP2 of the compare/capture channel.

4) The signal ETRF from the external clock pin input.

11.2.2.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE position is set, the external clock source mode 2 is used. when using the external clock source mode 2, ETRF is selected as CK_PSC. the ETR pin becomes ETRP after passing through the optional inverter (ETP), divider (ETPS), and then ETRF after passing through the filter (ETF).

With the ECE position bit and the SMS set to 111b, this is equivalent to the TS selecting ETRF as an input.

11.2.2.4 Encoder Mode

Setting the SMS to 001b, 010b, 011b will enable the encoder mode. Enabling encoder mode allows you to select a specific level in TI1FP1 and TI2FP2 to signal the output with another jump edge as the signal. This mode is used when an external encoder is used. Refer to Section 11.3.9 for specific functions.

11.2.3 Counters and Peripherals

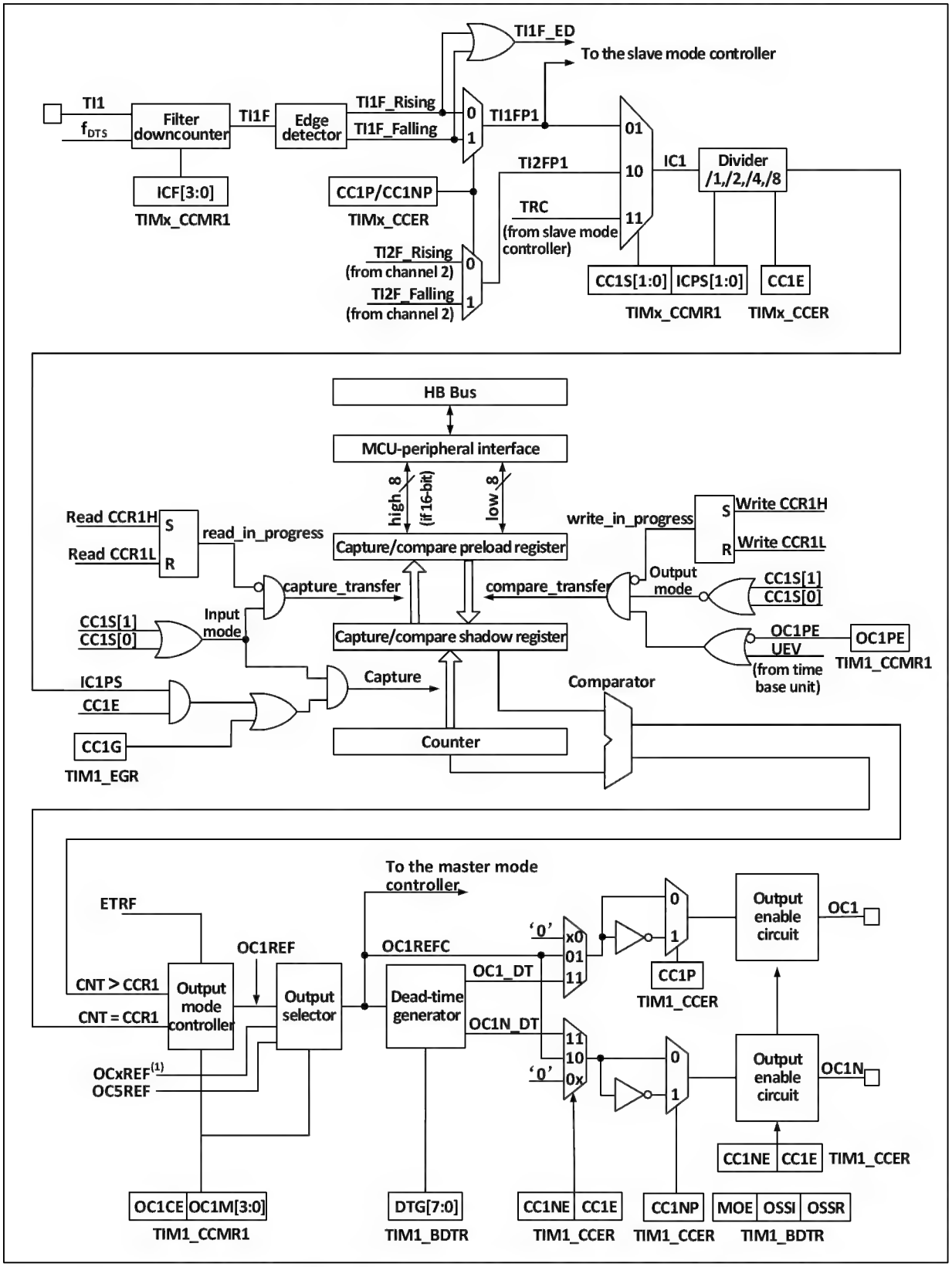
CK_PSC is input to the prescaler (PSC) for dividing. the PSC is 16-bit and the actual dividing factor is equal to the value of R16_TIMx_PSC + 1. CK_PSC goes through the PSC and becomes CK_INT. changing the value of R16_TIM1_PSC does not take effect in real time, but is updated to the PSC after an update event. the update event includes a UG bit clear and reset. The core of the timer is a 16-bit counter (CNT). CK_CNT is eventually fed to the CNT, which supports incremental count mode, decremental count mode, and incremental

and decremental count modes, and has an Automatic Reload Register (ATRLR) that reloads the initial value for the CNT at the end of each count cycle. There is also an auxiliary counter that keeps track of the number of times the ATRLR reloads the initial value for the CNT and can generate a specific event when the number of times set in the Repeat Count Register (RPTCR) is reached.

10.2.4 Compare/Capture Channels and Perimeters

The core of the timer is the compare/capture register, which is complemented by digital filtering, frequency division and inter-channel multiplexing in the peripheral input section, comparator and output control in the output section.

Figure 11-3 Block diagram of the structure of the compare/capture channel



The structure block diagram of the comparison capture channel is shown in Figure 10-3. The signal is input from the channel x pin and optionally made as TIx (the source of TI1 can be more than just CH1, see the structure block diagram of timer 11-1), TI1 is passed through the filter (ICF[3:0]) to generate TI1F, and then divided into TI1F_Rising and TI1F_Falling through the edge detector, these two signals are selected (CC1P) to generate TI1FP1,

TI1FP1 and TI2FP1 from channel 2 are sent together to CC1S to select to become IC1, which is sent to the comparison capture register after ICPS dividing.

The compare capture register consists of a preload register and a shadow register, and the read/write process operates only on the preload register. In capture mode, the capture occurs on the shadow register and is then copied to the preload register; in compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared to the core counter (CNT).

11.3 Function and Implementation

The implementation of the complex functions of the advanced-control timer are all achieved by the operation of the timer's compare/capture channel, clock input circuit and counter and peripheral parts. The clock input to the timer can come from multiple clock sources, including the input to the compare/capture channel. The operation of the compare/capture channel and clock source selection directly determines its function. The compare/capture channel is bidirectional and can operate in both input and output modes.

11.3.1 Input Capture Mode

The input capture mode is one of the basic functions of the timer. The principle of input capture mode is that a capture event occurs when a determined edge on the ICxPS signal is detected, and the current value of the counter is latched into the compare capture register (R16_TIMx_CHCTLRx). When a capture event occurs, CCxIF (in R16_TIMx_INTFR) is set, and if an interrupt or DMA is enabled, the corresponding interrupt or DMA is also generated. if CCxIF is already set when a capture event occurs, the CCxOF bit is set. CCxIF can be cleared by software, or by hardware by reading the compare capture register. CCxOF is cleared by software.

An example of channel 1 to illustrate the steps to use the input capture mode is as follows.

1) Configure the CCxS domain to select the source of the ICx signal. For example, set to 10b and select TI1FP1 as the source of IC1 instead of using the default setting, where the CCxS domain defaults to making the compare capture module the output channel.
2) Configure the ICxF domain to set the digital filter for the TI signal. The digital filter will sample the signal at a determined frequency, a determined number of times, and then output a hop. This sampling frequency and number of times is determined by ICxF.
3) Configure the CCxP bit to set the polarity of the TIxFPx. For example, keeping the CC1P bit low and selecting rising edge jumps.
4) Configure the ICxPS domain to set the ICx signal to be the crossover factor between ICxPS. For example, keeping ICxPS at 00b, without crossover.
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the compare capture register. Set the CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to determine whether to allow enable interrupts or DMA.

This completes the comparison capture channel configuration.

When a captured pulse is input to TI1, the value of the core counter (CNT) is recorded in the compare capture register, CC1IF is set, and the CCIOF bit is set when CC1IF has been set before. If the CC1IE bit is set, then an interrupt is generated; if CC1DE is set, a DMA request is generated. An input capture event can be generated by software by writing the event generation register (TIMx_SWEVGR).

11.3.2 Compare Output Mode

The compare output mode is one of the basic functions of the timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) agrees with the value of the compare

capture register. the OCxM field (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is a definite high or low level or a level flip. The CCxIF bit is also set when a compare coherent event is generated. If the CCxIE bit is pre-set, an interrupt will be generated; if the CCxDE bit is pre-set, a DMA request will be generated.

To configure to compare output modes, proceed as follows.

1) Configuring the clock source and auto-reload value of the core counter (CNT).
2) Setting the count value to be compared to the comparison capture register (R32_TIMx_CHxCVR).
3) If an interrupt needs to be generated, set the CCxIE bit.
4) Keeping OCxPE at 0 to disable the preload register of the compare register.
5) Setting the output mode, setting the OCxM field and the CCxP bit.
6) Enable the output, setting the CCxE bit.
7) Set the CEN bit to start the timer.

11.3.3 Forced Output Mode

The output pattern of the timer's compare capture channel can be forced by software to output a determined level without relying on comparison of the compare capture register's shadow register with the core counter.

This is done by setting OCxM to 100b, which forces OCxREF to low, or by setting OCxM to 101b, which forces OCxREF to high.

Note that by forcing OCxM to 100b or 101b, the comparison process of the internal core counters and compare capture registers is still going on, the corresponding flags are still set, and interrupts and DMA requests are still being generated.

11.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of PWM and is a special case of the input capture mode. The operation is the same as input capture mode except for the following differences: PWM occupies two compare capture channels and the input polarity of the two channels is set to opposite, one of the signals is set as trigger input and SMS is set to reset mode.

For example, to measure the period and frequency of the PWM wave input from TI1, the following operations are required.

1) Set TI1 (TI1FP1) to be the input of IC1 signal. Set CC1S to 01b.
2) Set TI1FP1 to rising edge active. Holding CC1P at 0.
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b.
4) Select TI1FP2 to set to falling edge active. Set CC2P to 1.
5) Select TI1FP1 as the source of the clock source. set TS to 101b.
6) Set the SMS to reset mode, i.e. 100b.
7) Enables input capture. CC1E and CC2E are set.

Thus the value of compare capture register 1 is the period of the PWM, and the value of compare capture register 2 is its duty cycle.

Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIM1_CH1/TIM1_CH2 can be used for PWM input mode.

11.3.5 PWM Output Mode

PWM output mode is one of the basic functions of the timer. PWM output mode is most commonly used to determine the PWM frequency using the reload value and the duty cycle using the capture comparison register. Set 110b or

111b in the OCxM field to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable automatic reload of the preload register. Since the value of the preload register can only be sent to the shadow register when an update event occurs, the UG bit needs to be set to initialize all registers before the core counter starts counting. In PWM mode, the core counter and the compare capture register are always comparing, and depending on the CMS bit, the timer is able to output edge-aligned or center-aligned PWM signals.

- Edge alignment

When edge alignment is used, the core counter is incremented or decremented, and in the PWM mode 1 scenario, OCxREF is high when the core counter value is greater than the compare capture register, and low when the core counter value is less than the compare capture register (e.g., when the core counter grows to the value of R16_TIMx_ATRLR and reverts to all zeros).

- Central alignment

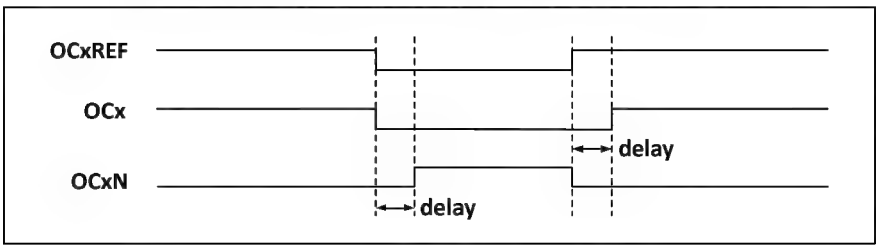
When using the central alignment modes, the core counter runs in alternating incremental and decremental count modes, and OCxREF makes rising and falling jumps when the values of the core counter and the compare capture register match. However, the comparison flags are set at different times in the three central alignment modes. When using the central alignment modes, it is best to generate a software update flag (Set the UG bit) before starting the core counter.

11.3.6 Complementary Output and Dead-time Insertion

The compare/capture channel generally has two output pins (Compare/capture channel 4 has only one output pin) and can output two complementary signals (OCx and OCxN). OCx and OCxN can be independently set for polarity via the CCxP and CCxNP bits, independently set for output enable via CCxE and CCxNE, and independently set for output enable via the MOE, OIS, OISN, OSSI, and OSSR bits for dead-time and other controls. Enabling the OCx and OCxN outputs simultaneously will insert a dead-time, and each channel has a 10-bit dead-time generator. OCx and OCxN are generated by the OCxREF association. If both OCx and OCxN are high active, then OCx is the same as OCxREF except that the rising edge of OCx is equivalent to OCxREF with a delay, and OCxN is the opposite of OCxREF in that its rising edge will have a delay relative to the falling edge of the reference signal. If the delay is greater than the effective output width, the corresponding pulse will not be generated.

The relationship between OCx and OCxN and OCxREF is illustrated in Figure 10-4, which shows the dead-time.

Figure 11-4 Complementary outputs and dead-time



11.3.7 Brake Signal

When the brake signal is generated, the output enable signal and invalid level are modified according to the MOE, OIS, OISN, OSSI, and OSSR bits. However, OCx and OCxN will not be at the active level at any time. The source of the brake event can come from the brake input pin or it can be a clock failure event which is generated by the CSS (Clock Safety System).

After system reset, the brake function is disabled by default (MOE bit is low), and setting the BKE bit enables the brake function. The polarity of the input brake signal can be set by setting BKP, and the BKE and BKP signals can be written at the same time, and there is a delay of one HB clock before the actual writing, so you need to wait for

one HB cycle to read the written value correctly.

At the presence of the selected level on the brake pin the system will generate the following actions.

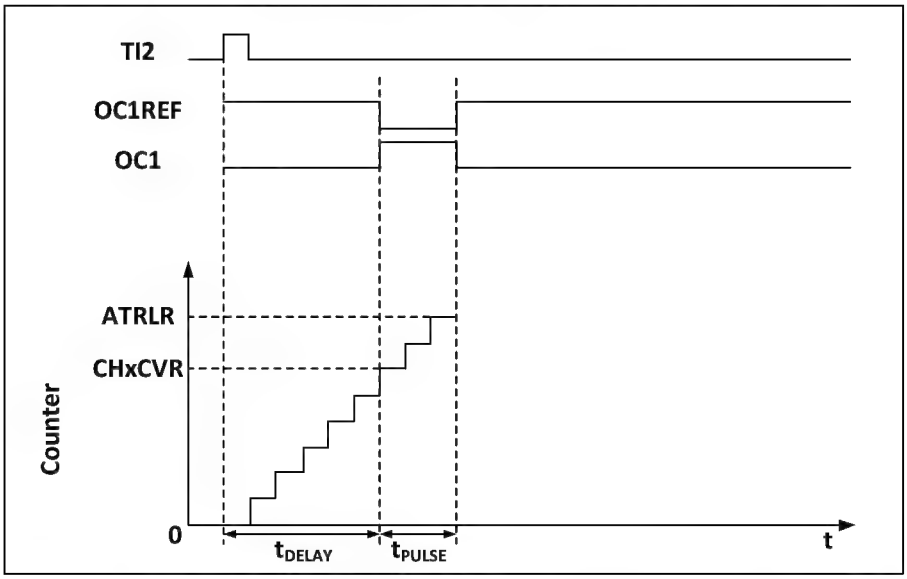
1) The MOE bit is cleared asynchronously, setting the output to an invalid, idle or reset state, depending on the setting of the SOOI bit.
2) After the MOE has been cleared, each output channel outputs a level determined by OISx.
3) When using complementary outputs: the outputs are placed in a null state, depending on the polarity.
4) If the BIE is set, an interrupt is generated when the BIF is set; if the BDE bit is set, a DMA request is generated.
5) If the AOE is set, the MOE bit is automatically set at the next update event UEV.

11.3.8 Single Pulse Mode

Single pulse mode can be used to allow the microcontroller to respond to a specific event by causing it to generate a pulse after a delay, with the delay and width of the pulse programmable. Placing the OPM bit allows the core counter to stop when the next update event UEV is generated (Counter flips to 0).

As shown in Figure 11-5, a positive pulse of length Tpulse needs to be generated on OC1 after a delay Tdelay at the beginning of a rising edge detected on the TI2 input pin.

Figure 11-5 Generation of single pulse



1) Set TI2 to trigger. Setting the CC2S field to 01b to map TI2FP2 to TI2; setting the CC2P bit to 0b to set TI2FP2 as rising edge detection; setting the TS field to 110b to set TI2FP2 as trigger source; setting the SMS field to 110b to set TI2FP2 to be used to start the counter.
2) Tdelay is determined by the value of the Compare Capture Register, and Tpulse is determined by the value of the Auto Reload Value Register and the Compare Capture Register.

11.3.9 Encoder Mode

The encoder mode is a typical application of the timer and can be used to access the biphasic output of the encoder. The counting direction of the core counter is synchronized with the direction of the encoder's rotation axis, and each pulse output from the encoder will cause the core counter to add or subtract one. To use the encoder, set the SMS field to 001b (Count only on TI2 edge), 010b (Count only on TI1 edge) or 011b (Count on both TI1 and TI2 edges), connect the encoder to the input of comparison capture channels 1 and 2, and set a value for the reload value register, which can be set to a larger value. When in encoder mode, the internal compare/capture register, prescaler, repeat count register, etc. of the timer are working normally. The following table shows the relationship between the

counting direction and the encoder signal.

Table 11-1 Relationship between counting direction and encoder signal of timer encoder mode

Counting effective edges	The level of relative signals	TI1FP1 signal edge		TI2FP2 signal	
		Rising edge	Falling edge	Rising edge	Falling edge
Counting at TI1 edge only	high	Downward counting	Upward counting	No count	
	low	Upward counting	Downward counting		
Counting at TI2 edge only	high	No count		Upward counting	Downward counting
	low			Downward counting	Upward counting
Double edge counting at TI1 and TI2	high	Downward counting	Upward counting	Upward counting	Downward counting
	low	Upward counting	Downward counting	Downward counting	Upward counting

11.3.10 Timer Synchronization Mode

Timers are capable of outputting clock pulses (TRGO) and receiving inputs from other timers (ITRx). The source of ITRx (TRGO from other timers) is different for different timers. The timer internal trigger connections are shown in Table 11-2.

Table 11-2 TIMx internal trigger connections

Slave timer	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TIM1	-	TIM2	-	-
TIM2	TIM1	-	-	-

11.3.11 Debug Mode

When the system enters debug mode, the timer continues to run or stops according to the settings of the DBG module.

11.4 Register Description

Table 11-3 TIM1-related registers list

Name	Access address	Description	Reset value
R16_TIM1_CTLR1	0x40012C00	Control Register 1	0x0000
R16_TIM1_CTLR2	0x40012C04	Control Register 2	0x0000
R16_TIM1_SMCFGR	0x40012C08	Slave Mode Control Register	0x0000
R16_TIM1_DMAINTENR	0x40012C0C	DMA/Interrupt Enable Register	0x0000
R16_TIM1_INTFR	0x40012C10	Interrupt Status Register	0x0000
R16_TIM1_SWEVGR	0x40012C14	Event Generation Register	0x0000

R16_TIM1_CHCTLR1	0x40012C18	Compare/Capture Control Register 1	0x0000
R16_TIM1_CHCTLR2	0x40012C1C	Compare/Capture Control Register 2	0x0000
R16_TIM1_CCER	0x40012C20	Compare/Capture Enable Register	0x0000
R16_TIM1_CNT	0x40012C24	Counter of Advanced-control Timer	0x0000
R16_TIM1_PSC	0x40012C28	Counting Clock Prescaler	0x0000
R16_TIM1_ATRLR	0x40012C2C	Auto-reload Value Register	0xFFFF
R16_TIM1_RPTCR	0x40012C30	Repeated Count Register	0x0000
R32_TIM1_CH1CVR	0x40012C34	Compare/Capture Register 1	0x00000000
R32_TIM1_CH2CVR	0x40012C38	Compare/Capture Register 2	0x00000000
R32_TIM1_CH3CVR	0x40012C3C	Compare/Capture Register 3	0x00000000
R32_TIM1_CH4CVR	0x40012C40	Compare/Capture Register 4	0x00000000
R16_TIM1_BDTR	0x40012C44	Brake and Dead-time Registers	0x0000
R16_TIM1_DMACFGR	0x40012C48	DMA Control Register	0x0000
R16_TIM1_DMAADR	0x40012C4C	DMA Address Register in Continuous Mode	0x0000

11.4.1 Control Register 1 (TIM1_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPLVL	CAPOV	Reserved				CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
15	CAPLVL	RW	Capture level indication enable in dual-edge capture. 1: Enable; 0: Disable. *Note: When enabled, [16] of CHxCVR indicates the level corresponding to the capture value.*	0
14	CAPOV	RW	Capture value mode configuration. 1: The CHxCVR value is 0xFFFF when a counter overflow is generated before capture. 0: The capture value is the value of the actual counter	0
[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These 2 bits define the division ratio between the timer clock (CK_INT) frequency, the dead-time and the sampling clock used by the dead-time generator and the digital filter (ETR, TIx). 00: Tdts=Tck_int; 01: Tdts = 2 × Tck_int; 10: Tdts = 4 × Tck_int; 11: Reserved.	0
7	ARPE	RW	Auto-reload preload enable bit. 1: Enable auto-reload of value register (ATRLR);	0

			0: Disable auto-reload of the value register (ATRLR).	
[6:5]	CMS[1:0]	RW	Central alignment mode selection. 00: Edge-aligned mode. The counter counts up or down based on the direction bit (DIR). 01: Central alignment mode 1. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down. 10: Central alignment mode 2. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up. 11: Central alignment mode 3. The counter counts up and down alternately. The output compare interrupt flag bit of the channel configured as output (CCxS=00 in the CHCTLRx register) is set when the counter counts both up and down. *Note: When the counter is enabled (CEN=1), the transition from edge-aligned mode to center-aligned mode is not allowed.*	0
4	DIR	RW	Counting direction. 1: The counting mode of the counter is minus counting; 0: The counting mode of the counter is increasing counting. *Note: This bit is not valid when the counter is configured in central alignment mode or encoder mode.*	0
3	OPM	RW	Single pulse mode. 1: The counter stops (clearing the CEN bit) when the next update event occurs. 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update request source, by which the software selects the source of the UEV event. 1: If an update interrupt or DMA request is enabled, only an update interrupt or DMA request is generated if the counter overflows/underflows. 0: If an update interrupt or DMA request is enabled, an update interrupt or DMA request is generated by any of the following events. -Counter overflow/underflow -Setting the UG position - Updates generated by the slave mode controlled	0

1	UDIS	RW	Disable updates, the software allows/disables the generation of UEV events by means of this bit. 1: UEV is disabled. no update event is generated and the registers (ARR, PSC, CHxCVR) keep their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counters and prescaler are reinitialized. 0: UEV is allowed. update (UEV) events are generated by any of the following events: -Counter overflow/underflow -Setting the UG position - Updates generated by the slave mode controlled Registers with caches are loaded with their preloaded values.	0
0	CEN	RW	Enable the counter. 1: Enable the counter. 0: Disable the counter. *Note: The external clock, gated mode and encoder mode will not work until the CEN bit is set in software. Trigger mode can automatically set the CEN bit in hardware.*	0

11.4.2 Control Register 2 (TIM1_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Reserved	CCPC

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	OIS4	RW	Output idle status 4. 1: When MOE=0, if OC4N is implemented, OC4=1 after dead-time; 0: When MOE=0, if OC4N is implemented, OC4=0 after dead-time. *Note: This bit cannot be modified after LOCK (TIM1_BDTR register) level 1, 2 or 3 has been set.*	0
13	OIS3N	RW	Output idle state 3. 1: when MOE = 0, OC3N = 1 after dead-time. 0: When MOE = 0, OC3N = 0 after dead-time. *Note: This bit cannot be modified after the LOCK (TIM1_BDTR register) level 1, 2 or 3 has been set.*	0
12	OIS3	RW	Output idle state 3, see OIS4.	0
11	OIS2N	RW	Output idle state 2, see OIS3N.	0

10	OIS2	RW	Output idle state 2, see OIS4.	0
9	OIS1N	RW	Output idle state 1, see OIS3N.	0
8	OIS1	RW	Output idle state 1, see OIS4.	0
7	TI1S	RW	TI1 selection. 1: TIM1_CH1, TIM1_CH2 and TIM1_CH3 pins connected to TI1 input after heterodyning. 0: TIM1_CH1 pin is connected directly to TI1 input.	0
[6:4]	MMS[2:0]	RW	Master mode selection: These 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. The possible combinations are as follows. 000: The UG bit of the Reset-TIM1_EGR register is used as the trigger output (TRGO). In the case of a reset generated by a trigger input (from a mode controller in reset mode), there is a delay in the signal on TRGO relative to the actual reset. 001: Enable - The counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or to control the enable from timers over a period of time. The counter enable signal is generated by the logical or of the trigger input signal in CEN control bit and gated mode. When the counter enable signal is controlled by a trigger input, there is a delay on TRGO unless master/slave mode is selected (see the description of the MSM bit in the TIM1_SMCR register). 010: Update - The update event is selected as a trigger input (TRGO). For example, the clock of a master timer may be used as a prescaler for a slave timer. 011: Compare pulse - on the occurrence of a capture or a successful comparison, when the CC1IF flag is to be set (even if it is already high), the trigger output sends a positive pulse (TRGO). 100: Compare - OC1REF signal is used as a trigger output (TRGO). 101: Compare - OC2REF signal is used as a trigger output (TRGO). 110: Compare - OC3REF signal is used as a trigger output (TRGO). 111: Compare - OC4REF signal is used as the trigger output (TRGO).	0
3	CCDS	RW	Capture the DMA selection for comparison. 1: Transmit a DMA request for CHxCVR when an update event occurs.	0

			0: Generate a DMA request for CHxCVR when CHxCVR occurs.	
2	CCUS	RW	Compare capture control update selection bits. 1: If CCPC is set, they can be updated by setting the COM bit or a rising edge on TRGI. 0: If the CCPC is set, they can only be updated by setting the COM bit. *Note: This bit only works for channels with complementary outputs.*	0
1	Reserved	RO	Reserved	0
0	CCPC	RW	Compare capture preload control bits. 1: The CCxE, CCxNE and OCxM bits are preloaded and when this bit is set they are only updated when the COM bit is set. 0: CCxE, CCxNE and OCxM bits are not preloaded. *Note: This bit only works for channels with complementary outputs.*	0

11.4.3 Slave Mode Control Register (TIM1_SMCFGR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR trigger polarity selection, this bit selects whether to input ETR directly or to input the inverse of ETR. 1: Invert ETR, active low or falling edge; 0: ETR, active high or rising edge.	0
14	ECE	RW	External clock mode 2 enable selection. 1: Enables external clock mode 2. 0: Disable external clock mode 2. *Note 1: Slave mode can be used simultaneously with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF in this case (TS bit cannot be '111').* *Note 2: When both external clock mode 1 and external clock mode 2 are enabled, the external clock input is ETRF.*	0
[13:12]	ETPS[1:0]	RW	External trigger prescaler, this signal frequency cannot exceed 1/4 of the TIM1CLK frequency at maximum, and can be down converted through this domain. 00: Prescaler off. 01: ETRP frequency divided by 2.	0

			10: ETRP frequency divided by 4. 11: ETRP frequency divided by 8.	
[11:8]	ETF[3:0]	RW	Externally triggered filtering, in fact, the digital filter is an event counter, which uses a certain sampling frequency to record up to N events and then produces a jump in the output. 0000: No filter, sampled in Fdts; 0001: Sampling frequency Fsampling=Fck_int, N=2. 0010: Sampling frequency Fsampling=Fck_int, N=4. 0011: Sampling frequency Fsampling=Fck_int, N=8. 0100: Sampling frequency Fsampling = Fdts/2, N = 6. 0101: Sampling frequency Fsampling = Fdts/2, N = 8. 0110: Sampling frequency Fsampling = Fdts/4, N = 6. 0111: Sampling frequency Fsampling = Fdts/4, N = 8. 1000: Sampling frequency Fsampling = Fdts/8, N = 6. 1001: Sampling frequency Fsampling = Fdts/8, N = 8. 1010: Sampling frequency Fsampling = Fdts/16, N = 5. 1011: Sampling frequency Fsampling = Fdts/16, N = 6. 1100: Sampling frequency Fsampling = Fdts/16, N = 8. 1101: Sampling frequency Fsampling = Fdts/32, N = 5. 1110: Sampling frequency Fsampling = Fdts/32, N = 6. 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0
7	MSM	RW	Master/slave mode selection. 1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is useful when the synchronization of several timers to a single external event is required. 0: Does not function.	0
[6:4]	TS[2:0]	RW	Trigger selection field, these 3 bits select the trigger input source used to synchronize the counter. 000: Internal trigger 0 (ITR0). 001: Internal trigger 1 (ITR1). 010: Internal trigger 2 (ITR2). 011: Internal trigger 3 (ITR3). 100: Edge detector of TI1 (TI1F_ED). 101: Filtered timer input 1 (TI1FP1). 110: Filtered timer input 2 (TI2FP2). 111: External trigger input (ETRF). The above only changes when SMS is 0. *Note: See Table 11-2 for details.*	0
3	Reserved	RO	Reserved	0
[2:0]	SMS[2:0]	RW	Input mode selection field. Selects the clock and trigger mode of the core counter.	0

			000: Driven by the internal clock CK_INT. 001: Encoder mode 1, where the core counter increments or decrements the count at the edge of TI2FP2 depending on the level of TI1FP1. 010: Encoder mode 2, where the core counter increments or decrements the count at the edge of TI1FP1, depending on the level of TI2FP2. 011: Encoder mode 3, where the core counter increments and decrements the count on the edges of TI1FP1 and TI2FP2 depending on the input level of another signal; 100: Reset mode, where the rising edge of the trigger input (TRGI) will initialize the counter and generate a signal to update the registers. 101: Gated mode, when the trigger input (TRGI) is high, the counter clock is turned on; at the trigger input becomes low, the counter is stopped, and the counter starts and stops are controlled. 110: Trigger mode, where the counter is started on the rising edge of the trigger input TRGI and only the start of the counter is controlled. 111: External clock mode 1, rising edge of the selected trigger input (TRGI) drives the counter.	

11.4.4 DMA/Interrupt Enable Register (TIM1_DMAINTENR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger DMA request enable bit. 1: Enable trigger DMA request; 0: Disable trigger DMA request.	0
13	COMDE	RW	COM's DMA request enable bit. 1: Enable COM's DMA request; 0: Disable COM's DMA request.	0
12	CC4DE	RW	DMA request enable bit of compare/capture channel 4. 1: Enable DMA request of compare/capture channel 4; 0: Disable DMA request of compare/capture channel 4.	0
11	CC3DE	RW	DMA request enable bit of compare/capture channel 3. 1: Enable DMA request of compare/capture channel 3; 0: Disable DMA request of compare/capture channel 3.	0
10	CC2DE	RW	DMA request enable bit of compare/capture channel 2.	0

			1: Enable DMA request of compare/capture channel 2; 0: Disable DMA request of compare/capture channel 2.	
9	CC1DE	RW	DMA request enable bit of compare/capture channel 1. 1: Enable DMA request of compare/capture channel 1; 0: Disable DMA request of compare/capture channel 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: Enable updated DMA request; 0: Disable updated DMA request.	0
7	BIE	RW	Brake interrupt enable bit. 1: Enable brake interrupt; 0: Disable brake interrupt.	0
6	TIE	RW	Trigger interrupt enable bit. 1: Enable trigger interrupt; 0: Disable trigger interrupt.	0
5	COMIE	RW	COM interrupt enable bit. 1: Enable COM interrupt; 0: Disable COM interrupt.	0
4	CC4IE	RW	Compare/capture channel 4 interrupt enable bit. 1: Enable compare/capture channel 4 interrupt; 0: Disable compare/capture channel 4 interrupt.	0
3	CC3IE	RW	Compare/capture channel 3 interrupt enable bit. 1: Enable compare/capture channel 3 interrupt; 0: Disable compare/capture channel 3 interrupt.	0
2	CC2IE	RW	Compare/capture channel 2 interrupt enable bit. 1: Enable compare/capture channel 2 interrupt; 0: Disable compare/capture channel 2 interrupt.	0
1	CC1IE	RW	Compare/capture channel 1 interrupt enable bit. 1: Enable compare/capture channel 1 interrupt; 0: Disable compare/capture channel 1 interrupt.	0
0	UIE	RW	Updated interrupt enable bit. 1: Enable updated interrupt; 0: Disable updated interrupt.	0

11.4.5 Interrupt Status Register (TIM1_INTFR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare/capture channel 4 repeated capture flag bit.	0
11	CC3OF	RW0	Compare/capture channel 3 repeated capture flag bit.	0
10	CC2OF	RW0	Compare/capture channel 2 repeated capture flag bit.	0

9	CC1OF	RW0	Compare/capture channel 1 repeated capture flag bit, only when the capture channel is configured to input capture mode. The flag is set by hardware and can be cleared by software write 0. 1: When the value of the counter is captured to the capture compare register, the state of the CC1IF has been set 0: There is no repetitive capture.	0
8	Reserved	RO	Reserved	0
7	BIF	RW0	Brake interrupt flag bit. Once the brake input is valid, the position can be zeroed by the hardware and cleared by the software. 1: Set effective level detected on brake pin input; 0: No brake event.	0
6	TIF	RW0	Trigger interrupt flag bit, when a trigger event occurs, the position bit is matched by the hardware and cleared by the software. The trigger event includes the detection of a valid edge at the TRGI input, or any edge in the gated mode, from a mode other than the gated mode. 1: Trigger event generation; 0: No trigger event generation.	0
5	COMIF	RW0	COM interrupt flag bit, once a COM event is generated, the bit is set by hardware and zeroed by software. COM events include CCxE, CCxNE, and OCxM being updated. 1: COM event generation; 0: No COM event generation.	0
4	CC4IF	RW0	Compare/capture channel 4 interrupt flag bit.	0
3	CC3IF	RW0	Compare/capture channel 3 interrupt flag bit.	0
2	CC2IF	RW0	Compare/capture channel 2 interrupt flag bit.	0
1	CC1IF	RW0	Compare/capture channel 1 interrupt flag bit. If the compare/capture channel is configured as output mode: This bit is set by hardware when the counter value matches the comparison value, except in centrosymmetric mode. The bit is cleared by the software. 1: The value of the core counter matches the value of compare/capture register 1; 0: No matching occurs. If compare/capture channel 1 is configured as input mode When the capture event occurs, the bit is set by the hardware, which is cleared by the software or by	0

			reading compare/capture register. 1: Counter is captured compare/capture register; 0: No input capture generation.	
0	UIF	RW0	Updated interrupt flag bit. When updated event occurs, the bit is set by hardware, cleared by software. 1: Updated event generation; 0: No updated event generation. The following feature will generate updated event: If UDIS=0, when the repeat counter value overflows or underflows; If URS=0, UDIS=0, when the UG bit is set, or when the counter core counter is reinitialized by software; If URS=0, UDIS=0, when the counter CNT is reinitialized by the triggered event.	0

11.4.6 Event Generation Register (TIM1_SWEVGR)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	BG	WO	Brake event generation bit. This Bit is set and cleared by software to generate a brake event. 1: There was a brake event. For MOE=0 and BIF=1, if the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA will be generated. 0: No action.	0
6	TG	WO	Trigger event generation bit. This bit is set by software and zeroed by hardware to generate a trigger event. 1: A trigger event is generated, and the TIF is set. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA are generated. 0: No action.	0
5	COMG	WO	Compare capture control update generation bits. Generates a compare/capture control update event. This bit is set by software and cleared automatically by hardware. 1: When CCPC=1, CCxE, CCxNE, OCxM bits are allowed to be updated; 0: No action. *Note: This bit is only valid for channels with complementary output (channel 1, 2, 3).*	0
4	CC4G	WO	Compare/capture event generation 4, generates	0

			compare/capture event 4.	
3	CC3G	WO	Compare/capture event generation 3, generates compare/capture event 3.	0
2	CC2G	WO	Compare/capture event generation 2, generates compare/capture event 2.	0
1	CC1G	WO	Compare/capture event generation 1, generates compare/capture event 1. This bit is set by software and zeroed by hardware. Used to generate a comparison capture event. 1: Generates a compare/capture event in compare/capture channel 1: If compare/capture channel 1 is configured as output: Set CC1IF bit. If enable corresponding interrupt and DMA, it will generate corresponding interrupt and DMA; If compare/capture channel 1 is configured as input: The current core counter' value is captured to compare/capture register 1; Set CC1IF bit, if enable corresponding interrupt and DMA, it will generate corresponding interrupt and DMA. If CC1IF is already set, set CC1OF bit. 0: No action.	G
0	UG	WO	Updated event generation bit, generates event. The bit is set by software, cleared by hardware. 1: Initialize the counter and generate an update event; 0: No action. *Note: The counter of the prescaler is also cleared, but the prescaler coefficient remains the same. If the core counter is cleared in centrosymmetric mode or increment mode, the core counter takes the value of the value register if it is in subtractive mode.*	0

11.4.7 Compare/Capture Control Register 1 (TIM1_CHCTLR1)

Offset address: 0x18

The channel can be used for input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bits. The functions of other bits of the register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (Pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare/capture channel 2 clear enable bit. 1: Once the ETRF input high level is detected, clear the OC2REF bit zero. 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare capture channel 2 mode settings fields. The three bits define the action of outputting the reference signal OC2REF, while OC2REF determines the values of OC2 and OC2N. OC2REF is efficient at high levels, while the effective levels for OC2 and OC2N depend on the CC2P and CC2NP bits. 000: Frozen. The comparison between the value of the compare/capture register and the core counter has no effect on OC2REF; 001: Force to be set to a valid level. Force OC2REF to be high when the core counter is the same as the value of compare/capture register 2; 010: Force to be set to an invalid level. Force OC2REF to be low when the value of the core counter is the same as that of compare/capture register 2 011: Flip. Flip the level of the OC2REF when the core counter is the same as the value of the compare/capture register 1; 100: Force to be set to an invalid level. Force OC2REF to be low; 101: Force to be set to a valid level. Force OC2REF to be high; 110: PWM Mode 1: In up count, once the core counter is less than the value of the compare capture register, channel 2 is valid level, otherwise it is invalid level; in downward counting, once the core counter is greater than the value of the compare capture register, channel 2 is invalid level (OC2REF=0), otherwise it is valid level (OC2REF=1); 111: PWM mode 2: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is invalid level, otherwise it is valid level; in down count, once the core counter is larger than the value of the compare capture register, channel 2 is valid level (OC2REF=1), otherwise it is invalid level (OC2REF=0). *Note: Once the LOCK level is set to 3 and CC2S=00b, this bit cannot be modified. In PWM mode 1 or PWM mode 2, the OC2REF level changes only when the*	0

			comparison result changes or when the output compare mode is switched from frozen mode to PWM mode.	
11	OC2PE	RW	Compare capture register 2 preload enable bit. 1: Enable preload function of the compare/capture register 2. The read and write operation only operates on the preload register. The preload value of the compare/capture register 2 is loaded into the current shadow register when the update event arrives. 0: Disable preload function of the compare/capture register 2. Compare/capture register 2 can be written at any time and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified; only in mono-pulse mode (OPM=1) you can use PWM mode without confirming the preload register, otherwise its action is uncertain.*	0
10	OC2FE	RW	Compare/capture channel 2 fast enable level. This bit is used to speed up the compare/capture channel of the response of output to triggered input events. 1: The input to the valid edge of the trigger acts as if a comparison match had occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the effective edge of the sampling trigger and the output of the compare/capture channel 2 is shortened to three clock cycles; 0: According to the value of the counter and the compare capture register 1, compare/capture channel 2 operate normally, even if the trigger is turned on. When the input of the flip-flop has a valid edge, the minimum delay of activating the output of the compare/capture channel 2 is 5 clock cycles. *Note: OC2FE works only if the channel is configured in PWM1 or PWM2 mode.*	0
[9:8]	CC2S[1:0]	RW	Compare/capture channel 2 input selection fields. 00: Compare/capture channel 2 is configured to output; 01: Compare/capture channel 2 is configured as input and IC2 is mapped on TI2; 10: Compare/capture channel 2 is configured as input and IC2 is mapped on TI1; 11: Compare/capture channel 2 is configured as input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit). *Note: Compare/capture channel 2 is writable only when*	0

			the channel is closed (CC2E is 00:00).	
7	OC1CE	RW	Compare/capture channel 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Compare/capture channel 1 mode setting fields.	0
3	OC1PE	RW	Compare/capture register 1 reload enable bit.	0
2	OC1FE	RW	Compare/capture channel 1 fast enable bit.	0
[1:0]	CC1S[1:0]	RW	Compare/capture channel 1 input selection fields.	0

Capture mode (Pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	Input capture filter 2 configuration domain. These bits set the sampling frequency of TI1 input and the length of digital filter. The digital filter consists of an event counter, which records N events and produces an output jump. 0000: No filter, sample as fDTS; 1000: Sampling frequency Fsampling=Fdts/8, N=6; 0001: Sampling frequency Fsampling=Fck_int, N=2; 1001: Sampling frequency Fsampling=Fdts/8, N=8; 0010: Sampling frequency Fsampling=Fck_int, N=4; 1010: Sampling frequency Fsampling=Fdts/16, N=5; 0011: Sampling frequency Fsampling=f=Fck_int, N=8; 1011: Sampling frequency Fsampling=Fdts/16, N=6; 0100: Sampling frequency Fsampling=Fdts/2, N=6; 1100: Sampling frequency Fsampling=Fdts/16, N=8; 0101: Sampling frequency Fsampling=Fdts/2, N=8; 1101: Sampling frequency Fsampling=Fdts/32, N=5; 0110: Sampling frequency Fsampling=Fdts/4, N=6; 1110: Sampling frequency Fsampling=Fdts/32, N=6; 0111: Sampling frequency Fsampling=Fdts/4, N=8; 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0
[11:10]	IC2PSC[1:0]	RW	Compare/capture channel 2 prescaler configuration domain. These two bits define the pre-division coefficient of the comparison capture channel 2. Once CC1E=0, the prescaler resets. 00: Without a prescaler, each edge detected on the capture input triggers a capture.; 01: Capture is triggered every 2 events; 10: Capture is triggered every 4 events; 11: : Capture is triggered every 8 events.	0
[9:8]	CC2S[1:0]	RW	Compare/capture channel 2 input selection domain, these two bits define the direction of the channel (input / output), and the selection of the input pin. 00: Compare/capture channel 1 channel is configured to output; 01: Compare/capture channel 1 channel is configured as	0

			input and IC1 is mapped on TI1; 10: Compare/capture channel 1 channel is configured as input and IC1 is mapped on TI2; 11: Compare/capture channel 1 channel is configured as input and the IC1 is mapped on the TRC. This mode works only when the internal trigger input is selected (selected by the TS bit). *Note: CC1S is writable only when the channel is closed (CC1E is 0).*	
[7:4]	IC1F[3:0]	RW	Input capture filter 1 configuration domain.	0
[3:2]	IC1PSC[1:0]	RW	Compare/capture channel 1 prescaler configuration domain.	0
[1:0]	CC1S[1:0]	RW	Compare/capture channel 1 input selection domain.	0

11.4.8 Compare/Capture Control Register 2 (TIM1_CHCTLR2)

Offset address: 0x1C

The channel can be used for input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bits. The functions of other bits of the register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14 13 12	11	10	9 8	7	6 5 4	3	2	1 0
OC4CE	OC4M[2:0]	OC4PE	OC4FE	CC4S[1:0]	OC3CE	OC3M[2:0]	OC3PE	OC3FE	CC3S[1:0]
IC4F[3:0]		IC4PSC[1:0]			IC3F[3:0]		IC3PSC[1:0]		

Compare mode (Pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare/capture channel 4 clear enable bit.	0
[14:12]	OC4M[2:0]	RW	Compare/capture channel 4 mode setting domain.	0
11	OC4PE	RW	Compare/capture register 4 reload enable bit.	0
10	OC4FE	RW	Compare/capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	Compare/capture channel 4 input selection domain.	0
7	OC3CE	RW	Compare/capture channel 3 clear enable bit.	0
[6:4]	OC3M[2:0]	RW	Compare/capture channel 3 mode setting domain.	0
3	OC3PE	RW	Compare/capture register 3 reload enable bit.	0
2	OC3FE	RW	Compare/capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	Compare/capture channel 3 input selection domain.	0

Capture mode (Pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture filter 4 configuration domain.	0
[11:10]	IC4PSC[1:0]	RW	Compare/capture channel 4 prescaler configuration domain.	0
[9:8]	CC4S[1:0]	RW	Compare/capture channel 4 input selection domain	0

[7:4]	IC3F[3:0]	RW	Input capture filter 3 configuration domain.	0
[3:2]	IC3PSC[1:0]	RW	Compare/capture channel 3 prescaler configuration domain.	0
[1:0]	CC3S[1:0]	RW	Compare/capture channel 3 input selection domain	0

11.4.9 Compare/Capture Enable Register 2 (TIM1_CCER)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare/capture channel 4 output polarity setting bit.	0
12	CC4E	RW	Compare/capture channel 4 output enable bit.	0
11	CC3NP	RW	Compare/capture channel 3 complementary output polarity setting bit.	0
10	CC3NE	RW	Compare/capture channel 3 complementary output enable bit.	0
9	CC3P	RW	Compare/capture channel 3 output polarity setting bit.	0
8	CC3E	RW	Compare/capture channel 3 output enable bit.	0
7	CC2NP	RW	Compare/capture channel 2 complementary output polarity setting bit.	0
6	CC2NE	RW	Compare/capture channel 2 complementary output enable bit.	0
5	CC2P	RW	Compare/capture channel 2 output polarity setting bit.	0
4	CC2E	RW	Compare/capture channel 2 output enable bit.	0
3	CC1NP	RW	Compare/capture channel 1 complementary output polarity setting bit.	0
2	CC1NE	RW	Compare/capture channel 1 complementary output enable bit.	0
1	CC1P	RW	Compare/capture channel 1 output polarity setting bit. CC1 channel is configured as output: 1: OC1 active low; 0: OC1 active high. CC1 channel is configured as input: The bit selects whether the inverse signal of IC1 or IC1 is used as the trigger or capture signal. 1: Invert: capture occurs at the falling edge of IC1; when used as an external trigger, IC1 inverts 0: No inversion: capture occurs on the rising edge of the IC1; when used as an external trigger, the IC1 is not inverted. *Note: Once the LOCK level (the LOCK bit in the*	0

			TIM1_BDTR register) is set to 3 or 2, the bit cannot be modified.	
0	CC1E	RW	Compare/capture channel 1 output enable bit CC1 channel is configured as output: 1: Enable. The OC1 signal is output to the corresponding output pin, and its output level depends on the values of MOE, OSSI, OSSR, OIS1, OIS1N and CC1NE bits. 0: Disable. OC1 disables output, so the output level of OC1 depends on the values of the MOE, OSSI, OSSR, OIS1, OIS1N, and CC1NE bits. CC1 channel is configured as input: This bit determines whether the value of the counter can be captured into the TIM1_CH1CVR register. 1: Capture enabled; 0: Capture disabled.	0

11.4.10 Counter for Advanced-control Timer (TIM1_CNT)

Offset address: 0x24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the counter for the timer.	0

11.4.11 Counting Clock Prescaler (TIM1_PSC)

Offset address: 0x28

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The frequency division coefficient of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the frequency divider / (PSC+1).	0

11.4.12 Auto-reload Value Register (TIM1_ATRLR)

Offset address: 0x2C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATRLR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	ATRLR[15:0]	RW	The value of this field will be loaded into the counter.	FFFFh

			When ATRLR acts and updates, see Section 11.2.3; when ATRLR is empty, the counter stops.	

11.4.13 Repeat Count Value Register (TIM1_RPTCR)

Offset address: 0x30

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								RPTCR[7:0]							

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:0]	RPTCR[7:0]	RW	The value of repeated counter.	0

11.4.14 Compare/Capture Register 1 (TIM1_CH1CVR)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL1	RO	The level indication bit corresponding to the captured value.	0
[15:0]	CH1CVR[15:0]	RW	The value of compare/capture register channel 1.	0

11.4.15 Compare/Capture Register 2 (TIM1_CH2CVR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL2	RO	The level indication bit corresponding to the captured value.	0
[15:0]	CH2CVR[15:0]	RW	The value of compare/capture register channel 2.	0

11.4.16 Compare/Capture Register 3 (TIM1_CH3CVR)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL3	RO	The level indication bit corresponding to the captured value.	0
[15:0]	CH3CVR[15:0]	RW	The value of compare/capture register channel 3.	0

11.4.17 Compare/Capture Register 4 (TIM1_CH4CVR)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL4	RO	The level indication bit corresponding to the captured value.	0
[15:0]	CH4CVR[15:0]	RW	The value of compare/capture register channel 4.	0

11.4.18 Brake and Dead-time Register (TIM1_BDTR)

Offset address: 0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							

Bit	Name	Access	Description	Reset value
15	MOE	RW	The main output enable bit. Once the brake signal is valid, it will be cleared asynchronously. 1: Enable OCx and OCxN to be set as output; 0: Disable output of OCx and OCxN or force it to be idle	0
14	AOE	RW	Automatic output enable. 1: MOE can be set by the software or in the next update event;	0

			0: MOE can only be set by software.	
13	BKP	RW	Brake input polarity setting bit. 1: Brake input active high; 0: Brake input active low. *Note: When LOCK level 1 is set, this bit cannot be modified. Writing to this bit requires an HB clock to take effect.*	0
12	BKE	RW	Brake function enable bit. 1: Enable brake input; 0: Disable brake input. *Note: When LOCK level 1 is set, this bit cannot be modified. Writing to this bit requires an HB clock to take effect.*	0
11	OSSR	RW	1: When the timer does not work, once CCxE=1 or CCxNE=1, first turn on OC/OCN and output invalid level, then set OCx and OCxN to enable output signal = 1; 0: Disable OC/OCN output when the timer is not working. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
10	OSSI	RW	1: When the timer does not work, once CCxE=1 or CCxNE=1, OC/OCN first outputs its idle level, and then OCx, OCxN enable output signal = 1; 0: Disable OC/OCN output when the timer is not working. *Note: When LOCK level 1 is set, this bit cannot be modified.*	0
[9:8]	LOCK[1:0]	RW	Lock function settings domain. 00: Disable the locking function; 01: Lock level 1, cannot write DTG, BKE, BKP, AOE, OISx and OISxN bits; 10: Lock level 2, cannot write to the bits in lock level 1, nor can you write to CC polar bits and OSSR and OSSI bits; 11: Lock level 3, cannot write to the bits in lock level 2, nor can you write to CC control bits. *Note: After the system is reset, the LOCK bit can only be written once and cannot be modified again until it is reset.*	0

[7:0]	DTG[7:0]	RW	Dead-time setting bits, which define the dead-time duration between complementary outputs. Suppose DT indicates its duration: DTG[7:5]=0xx=>DT=DTG[7:0]*Tdtg, Tdtg =TDTS; DTG[7:5]=10x=>DT=（64+DTG[5:0]）*Tdtg，Tdtg=2*TDTS; DTG[7:5]=110=>DT=（32+DTG[4:0]）*Tdtg，Tdtg=8×TDTS; DTG[7:5]=111=>DT=（32+DTG[4:0]）*Tdtg，Tdtg=16 *TDTS.	0

11.4.19 DMA Control Register (TIM1_DMACFGR)

Offset address: 0x48

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DBL[4:0]					Reserved			DBA[4:0]				

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	The length of the continuous transmission of DMA. The actual value is the value of this field + 1.	0
[7:5]	Reserved	RO	Reserved	0
[4:0]	DBA[4:0]	RW	These bits define the offset of DMA from the address of control register 1 in continuous mode.	0

11.4.20 DMA Address Register in Continuous Mode (TIM1_DMAADR)

Offset address: 0x4C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAB[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	DMAB[15:0]	RW	The address of the DMA in continuous mode.	0

Chapter 12 General-purpose Timer (GPTM)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The general-purpose timer module contains a 16-bit timer TIM2 that can be reinstalled automatically, which is used to measure the pulse width or generate pulses and PWM waves of a specific frequency. It can be used in automatic control, power supply and other fields.

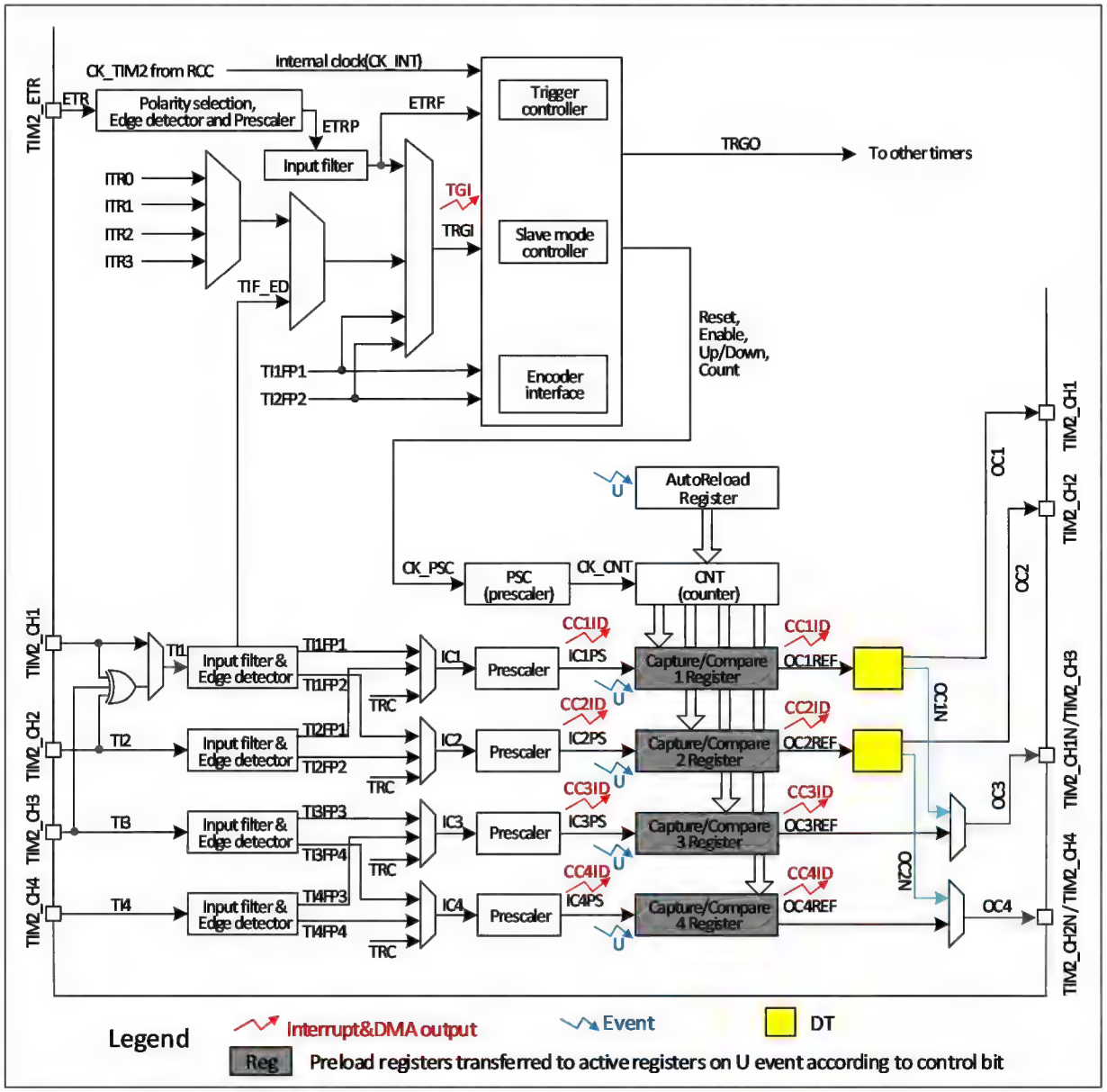
12.1 Main Features

The main features of the general-purpose timer include.

- 16-bit auto-reload counter, supports incremental counting mode, decremental counting mode and incremental and decremental counting mode
- 16-bit prescaler with dynamically adjustable crossover factor from 1 to 65536
- Support four independent comparison capture channels
- Each comparison capture channel supports multiple operating modes, such as: input capture, output comparison, PWM generation, and single pulse output
- Support external signal control timer
- Support DMA in multiple modes
- Support incremental coding, cascading and synchronization between timers

12.2 Principle and Structure

Figure 12-1 Block diagram of the structure of the general-purpose timer



12.2.1 Overview

As shown in Figure 12-1, the structure of the general-purpose timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare/capture channel part.

The clock of the general-purpose timer can come from the HB bus clock (CK_INT), from the external clock input pin (TIM2_ETR), from other timers with clock output function (ITRx), or from the input of the compare/capture channel (TIM2_CHx). After various set filtering and frequency division operations, these input clock signals become CK_PSC clocks and output to the core counter. In addition, these complex clock sources can also be output as TRGO to other timers and ADC peripherals.

The core of the general-purpose timer is a 16-bit counter (CNT). After the CK_PSC is divided by the prescaler (PSC), the CK_CNT is finally output to the CNT, CNT to support the increasing counting mode, the decreasing

counting mode and the increasing or decreasing counting mode, and there is an automatic reload value register (ATRLR) to reload the initialization values for the CNT at the end of each counting cycle.

The general-purpose timer has four sets of compare/capture channels, each of which can input pulses from exclusive pins or output waveforms to pins, i.e., the compare/capture channels support both input and output modes. The input of each channel of the compare/capture register supports filtering, dividing, edge detection, and other operations, and supports mutual triggering between channels, and can also provide clock for the core counter CNT. Each compare/capture channel has a set of compare/capture registers (CHxCVR) that support comparison with the main counter (CNT) to output pulses.

12.2.2 Difference between General-purpose Timer and Advanced-control Timer

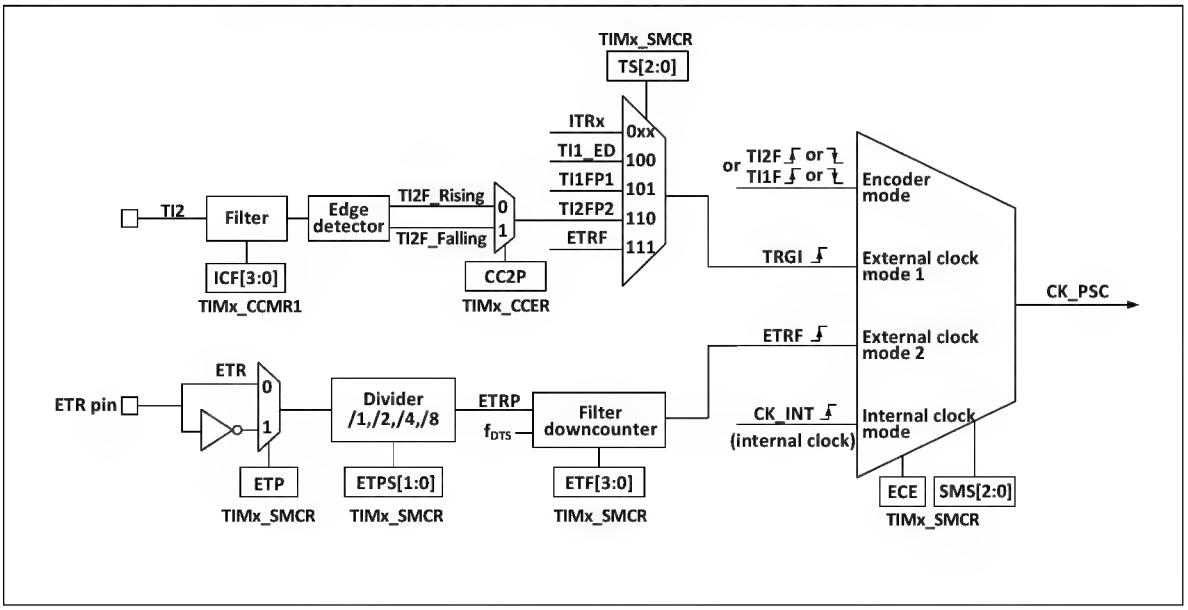
Compared to advanced-control timers, general purpose timers lack the following features.

1) The general-purpose timer lacks a repeat count register for counting the count cycles of the core counter.
2) The compare/capture channel of the general-purpose timer lacks dead-time generation and has no complementary output.
3) The general-purpose timer does not have a brake signal mechanism.

12.2.3 Clock Input

This section discusses the source of CK_PSC. The clock source portion of the overall block diagram of the general-purpose timer is captured here.

Figure 12-2 General-purpose Timer CK_PSC Source Block Diagram



The optional input clocks can be divided into 4 categories.

1) Route of the external clock pin (ETR) input: ETR → ETRP → ETRF.
2) Internal HB clock input route: CK_INT.
3) Route from the comparison capture channel pin (TIM2_CHx): TIMx_CHx → TIx → TIxFPx, this route is also used in encoder mode.
4) Input from other internal timers: ITRx.

The actual operation can be divided into 3 categories by determining the choice of input pulse for the SMS of the

CK_PSC source.

1) Selection of the internal clock source (CK_INT).
2) External clock source mode 1.
3) External clock source mode 2.
4) Encoder mode.

All 4 clock source sources mentioned above can be selected by these 4 operations.

12.2.3.1 Internal Clock Source (CK_INT)

If the general-purpose timer is started when the SMS field is held at 000b, then it is the internal clock source (CK_INT) that is selected as the clock. At this point CK_INT is CK_PSC.

12.2.3.2 External Clock Source Mode 1

When the SMS domain is set to 111b, external clock source mode 1 is enabled. When external clock source 1 is enabled, TRGI is selected as the source for CK_PSC. it is worth noting that the user also needs to select the source for TRGI by configuring the TS domain. the TS domain can select the following types of pulses as clock sources.

1) Internal trigger (ITRx, x is 0,1,2,3).
2) The signal after compare/capture channel 1 through the edge detector (TI1F_ED).
3) Signals TI1FP1, TI2FP2 of the compare/capture channel.
4) The signal ETRF from the external clock pin input.

12.2.3.3 External Clock Source Mode 2

Use external trigger mode 2 to count on every rising or falling edge of the external clock pin input. When the ECE position is set, the external clock source mode 2 is used. when using the external clock source mode 2, ETRF is selected as CK_PSC. the ETR pin becomes ETRP after passing through the optional inverter (ETP), divider (ETPS), and then ETRF after passing through the filter (ETF).

With the ECE position bit and the SMS set to 111b, then it is equivalent to the TS selecting ETRF as the input.

12.2.3.4 Encoder Mode

Setting the SMS to 001b, 010b, 011b will enable the encoder mode. Enabling encoder mode allows you to select a specific level in TI1FP1 and TI2FP2 to signal the output with another jump edge as the signal. This mode is used when an external encoder is used. Refer to Section 12.3.7 for specific functions.

12.2.4 Counters and Peripherals

The CK_PSC is input to the prescaler (PSC) for frequency division. PSC is 16-bit, and the actual frequency division coefficient is equivalent to the value of R16_TIM2_PSC + 1. CK_PSC becomes CK_INT after PSC. Changing the value of R16_TIM1_PSC does not take effect in real time, but is updated to PSC after the update event. Update events include UG bit zeroing and reset.

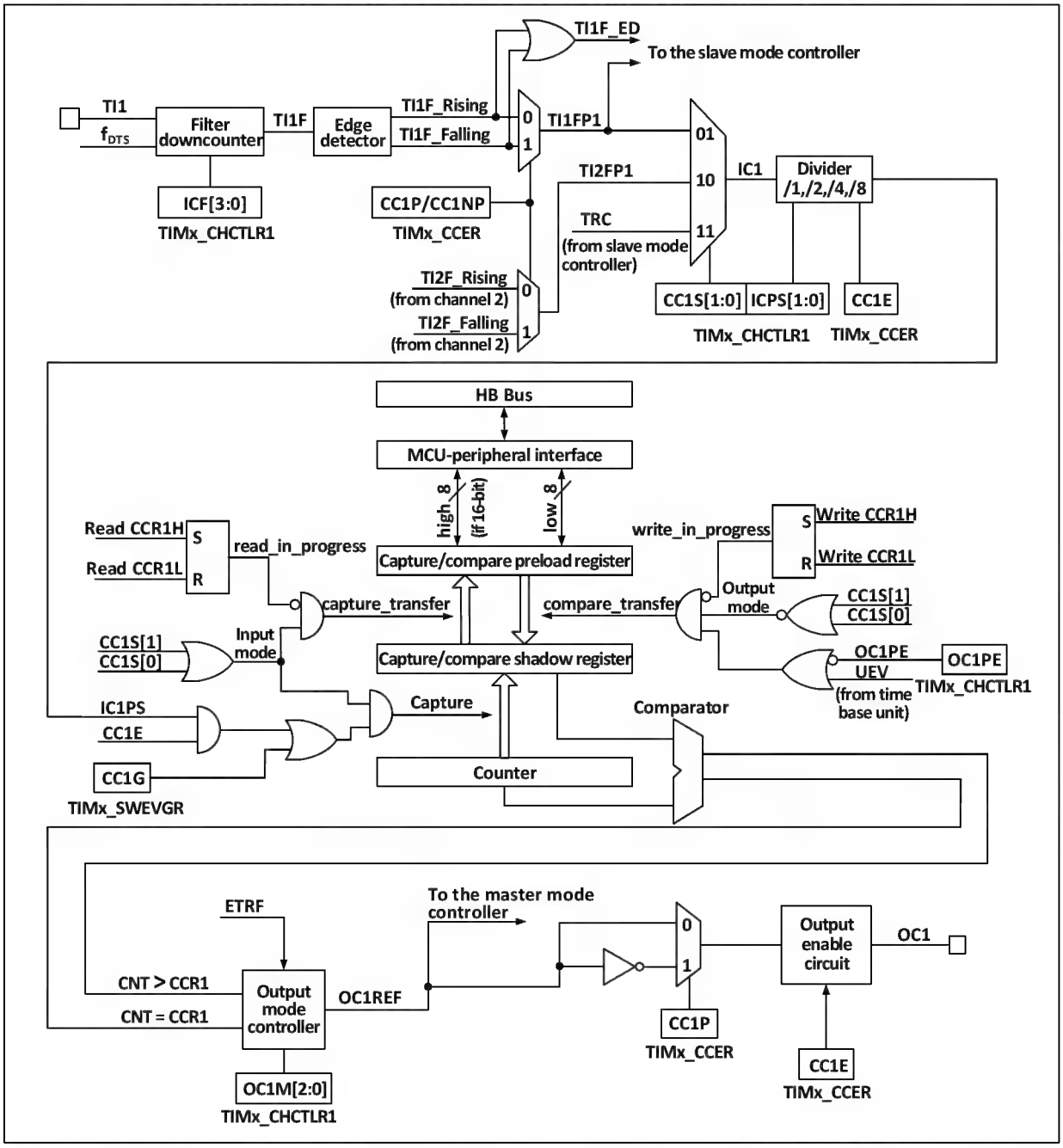
The core of the timer is a 16-bit counter (CNT). The CK_CNT will eventually be entered into the CNT, CNT to support the increase count mode, the decrease count mode, and the increase and decrease count mode, and there is an automatic reload value register (ATRLR) to reload the initial values for the CNT at the end of each count cycle.

12.2.5 Compare/Capture Channels

The core of the compare/capture channel, which is the core of the timer to achieve complex functions, is the compare/capture register, supplemented by digital filtering, frequency division and inter-channel multiplexing in

the peripheral input section, and comparator and output control in the output section. The structure block diagram of the compare/capture channel is shown in Figure 12-3.

Figure 12-3 Block diagram of the structure of the compare/capture channel



Signal input from the channel x pin can be selected as TIx (the source of TI1 can be more than just CH1, see timer block figure 121), TI1 through the filter (ICF [3:0]) to generate TI1F, and then after the edge detector is divided into TI1F_Rising and TI1F_Falling, the two signals after selection (CC1P) generated TI1FP1,TI1FP1 and TI2FP1 from channel 2 together to the CC1S to choose to become IC1, after ICPS frequency division to be sent to the comparison capture register.

The compare/capture register consists of a preload register and a shadow register, and the read and write process operates only the preload register. In capture mode, capture occurs on the shadow register and then copied to the preload register; in compare mode, the contents of the preload register are copied to the shadow register, and then the contents of the shadow register are compared with the core counter (CNT).

12.3 Function and Implementation

The complex functions of a general-purpose timer are implemented by manipulating the timer's compare/capture channel, clock input circuitry, and counter and peripheral components. The clock input to the timer can be derived from multiple clock sources including the input to the compare capture channel. The operation of the compare/capture host channel and clock source selection directly determines its function. The compare/capture channel is bidirectional and can operate in both input and output modes.

12.3.1 Input Capture Mode

The input capture mode is one of the basic functions of the timer. The principle of input capture mode is that when a determined edge on the ICxPS signal is detected, a capture event is generated and the current value of the counter is latched into the compare capture register (R16_TIM2_CHCTLRx). The CCxIF (in R16_TIM2_INTFR) is set when a capture event occurs, and the corresponding interrupt or DMA is generated if enabled. If the CCxIF is already set when a capture event occurs, the CCxOF bit is set. the CCxIF can be cleared by software, or by hardware by reading the compare/capture register. CCxOF is cleared by software.

An example of channel 1 to illustrate the steps to use the input capture mode is as follows.

1) Configure the CCxS domain to select the source of the ICx signal. For example, set it to 10b and select TI1FP1 as the source of IC1, not using the default setting, the CCxS domain defaults to making the comparison capture module the output channel.
2) Configure the ICxF domain to set the digital filter for the TI signal. The digital filter will sample the signal at a determined frequency, a determined number of times, and then output a hop. This sampling frequency and number of times is determined by ICxF.
3) Configure the CCxP bit to set the polarity of the TIxFPx. For example, keeping the CC1P bit low and selecting rising edge jumps.
4) Configure the ICxPS domain to set the ICx signal to be the crossover factor between ICxPS. For example, keeping ICxPS at 00b, without crossover.
5) Configure the CCxE bit to allow capturing the value of the core counter (CNT) into the compare capture register. Set the CC1E bit.
6) Configure the CCxIE and CCxDE bits as needed to determine whether to allow enable interrupts or DMA.

This completes the compare/capture channel configuration.

When a captured pulse is input to TI1, the value of the core counter (CNT) is recorded in the compare/capture register, CC1IF is set, and the CCIOF bit is set when CC1IF has been set before. If the CC1IE bit is set, then an interrupt is generated; if CC1DE is set, a DMA request is generated. An input capture event can be generated by software by way of writing the event generation register (R16_TIM2_SWEVGR).

12.3.2 Compare Output Mode

The compare output mode is one of the basic functions of the timer. The principle of the compare output mode is to output a specific change or waveform when the value of the core counter (CNT) agrees with the value of the compare/capture register. the OCxM field (in R16_TIMx_CHCTLRx) and the CCxP bit (in R16_TIMx_CCER) determine whether the output is a definite high or low level or a level flip. The CCxIF bit is also set when a compare coherent event is generated. If the CCxIE bit is pre-set, an interrupt will be generated; if the CCxDE bit is pre-set, a DMA request will be generated.

To configure to compare output modes, proceed as follows.

1) Configuring the clock source and auto-reload value of the core counter (CNT).

2) Set the count value to be compared to the compare/capture register (R16_TIM2_CHxCVR).
3) Set the CCxIE bit if an interrupt needs to be generated.
4) Keep OCxPE at 0 to disable the preload register for the compare/capture register.
5) Setting the output mode, setting the OCxM field and the CCxP bit.
6) Enable the output, setting the CCxE bit.
7) Setting the CEN bit to start the timer.

12.3.3 Forced Output Mode

The output pattern of the timer's compare capture channel can be forced by software to output a determined level without relying on comparison of the compare capture register's shadow register with the core counter.

This is done by setting OCxM to 100b, which forces OCxREF to low, or by setting OCxM to 101b, which forces OCxREF to high.

Note that by forcing OCxM to 100b or 101b, the comparison process between the internal main counter and the compare capture register is still going on, the corresponding flags are still set, and interrupts and DMA requests are still being generated.

12.3.4 PWM Input Mode

The PWM input mode is used to measure the duty cycle and frequency of PWM and is a special case of the input capture mode. The operation is the same as input capture mode except for the following differences: PWM occupies two compare/capture channels and the input polarity of the two channels is set to opposite, one of the signals is set as trigger input and SMS is set to reset mode.

For example, to measure the period and frequency of the PWM wave input from TI1, the following operations are required.

1) Set TI1 (TI1FP1) to be the input of IC1 signal. Set CC1S to 01b.
2) Set TI1FP1 to rising edge active. Holding CC1P at 0.
3) Set TI1 (TI1FP2) as the input of IC2 signal. Set CC2S to 10b.
4) Select TI1FP2 to set to falling edge active. Set CC2P to 1.
5) Select TI1FP1 as the source of the clock source. set TS to 101b.
6) Set the SMS to reset mode, i.e. 100b.
7) Enable input capture. CC1E and CC2E are set.

Note: Since only TI1FP1 and TI2FP2 are connected to the slave mode controller, only TIMx_CH1/TIMx_CH2 can be used for PWM input mode.

12.3.5 PWM Output Mode

PWM output mode is one of the basic functions of the timer. PWM output mode is most commonly used to determine the PWM frequency using the reload value and the duty cycle using the compare/capture register. Set 110b or 111b in the OCxM field to use PWM mode 1 or mode 2, set the OCxPE bit to enable the preload register, and finally set the ARPE bit to enable the automatic reload of the preload register. The value of the preload register can only be sent to the shadow register when an update event occurs, so the UG bit needs to be set to initialize all registers before the core counter starts counting. In PWM mode, the core counter and the compare/capture register are always comparing, and depending on the CMS bit, the timer is able to output edge-aligned or center-aligned PWM signals.

- Edge alignment

When using edge alignment, the core counter is incremented or decremented, and in the PWM mode 1 scenario, OCxREF rises to high when the core counter value is greater than the compare/capture register; when the core

counter value is less than the compare/capture register (For example, when the core counter grows to the value of R16_TIM2_ATRLR and reverts to full 0), OCxREF falls to low.

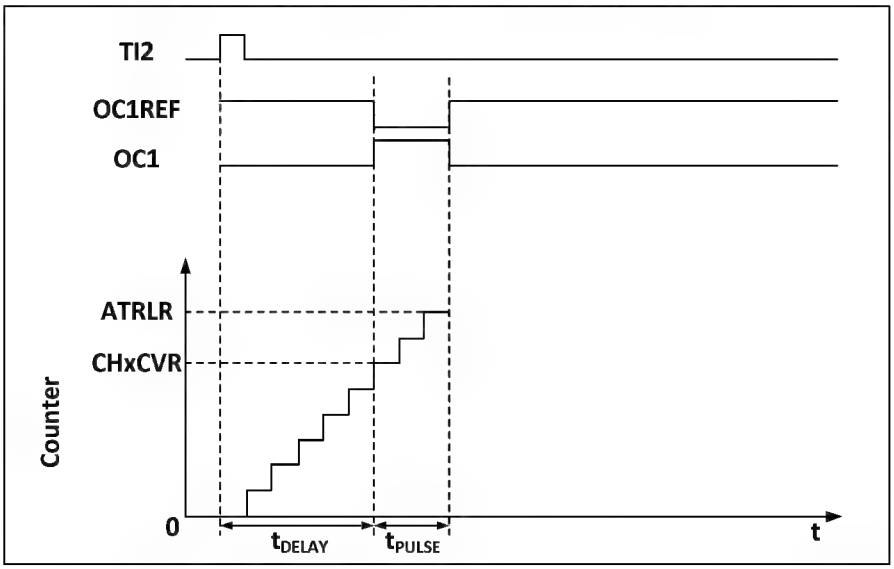
- Central alignment

When using the central alignment modes, the core counter runs in alternating incremental and decremental count modes, and OCxREF performs rising and falling jumps when the values of the core counter and the compare/capture register match. However, the comparison flags are set at different times in the three central alignment modes. When using the central alignment modes, it is best to generate a software update flag (set the UG bit) before starting the core counter.

12.3.6 Single Pulse Mode

The single pulse mode can respond to a specific event by generating a pulse after a delay, with programmable delay and pulse width. Setting the OPM bit stops the core counter when the next update event UEV is generated (counter flips to 0).

Figure 12-4 Event generation and impulse response



As shown in Figure 12-4, a positive pulse of length Tpulse needs to be generated on OC1 after a delay Tdelay at the beginning of a rising edge detected on the TI2 input pin.

1) Set TI2 to trigger. Setting the CC2S field to 01b to map TI2FP2 to TI2; setting the CC2P bit to 0b to set TI2FP2 as rising edge detection; setting the TS field to 110b to set TI2FP2 as trigger source; setting the SMS field to 110b to set TI2FP2 to be used to start the counter.
2) Tdelay is defined by the Compare/Capture Register and Tpulse is determined by the value of the Auto Reload Value Register and the Compare/Capture Register.

12.3.7 Encoder Mode

The encoder mode is a typical application of the timer and can be used to access the biphasic output of the encoder. The counting direction of the core counter is synchronized with the direction of the encoder's rotation axis, and each pulse output from the encoder will add or subtract one from the core counter. To use the encoder, set the SMS field to 001b (count only on TI2 edge), 010b (count only on TI1 edge) or 011b (count on both TI1 and TI2 edges), connect the encoder to the input of the compare/capture channels 1 and 2, and set a reload value counter value, which can be set to a larger value. When in encoder mode, the internal compare/capture register, prescaler, repeat count register, etc. of the timer are working normally. The following table shows the relationship between the counting direction

and the encoder signal.

Table 12-1 Relationship between counting direction and encoder signal of timer encoder mode

Counting effective edges	The level of relative signals	TI1FP1 signal edge		TI2FP2 signal edge	
		Rising edge	Falling edge	Rising edge	Falling edge
Counting at TI1 edge only	High	Downward counting	Upward counting	No count	
	Low	Upward counting	Downward counting		
Counting at TI2 edge only	High	No count		Upward counting	Downward counting
	Low			Downward counting	Upward counting
Double edge counting at TI1 and TI2	High	Downward counting	Upward counting	Upward counting	Downward counting
	Low	Upward counting	Downward counting	Downward counting	Upward counting

12.3.8 Timer Synchronization Mode

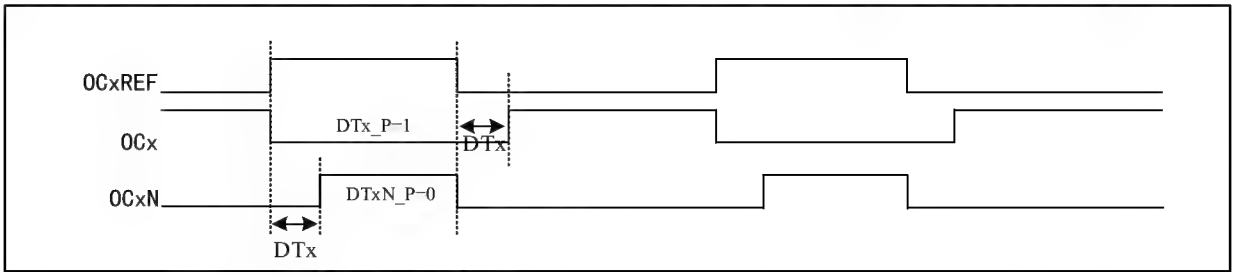
Timers are capable of outputting clock pulses (TRGO) and also receiving inputs from other timers (ITRx). The source of ITRx (TRGO from other timers) is different for different timers. The timer internal trigger connections are shown in Table 11-2.

Table 12-2 GTPM internal trigger connection

Slave timer	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TIM2	TIM1	-	-	-
TIM1	-	TIM2	-	-

12.3.9 Complementary Outputs and Deadtime

TIM2 is capable of outputting 2 complementary signals with dead-time (OCx and OCxN), which are independently polarity-controlled by the DTx_P and DTxN_P bits, and the size of the dead-time can be set by setting the DTx bit.



When complementary outputs are used, channels 3 and 4 act as channel 1 and channel 2 complementary output channels.

12.3.10 Debug Mode

When the system enters the debug mode, the timer can be controlled to continue running or stop according to the

setting of DBG module.

12.4 Register Description

Table 12-3 TIM2-related registers list

Name	Offset address	Description	Reset value
R16_TIM2_CTLR1	0x40000000	Control Register 1	0x0000
R16_TIM2_CTLR2	0x40000004	Control Register 2	0x0000
R16_TIM2_SMCFGR	0x40000008	Slave Mode Control Register	0x0000
R16_TIM2_DMAINTENR	0x4000000C	DMA/Interrupt Enable Register	0x0000
R16_TIM2_INTFR	0x40000010	Interrupt Status Register	0x0000
R16_TIM2_SWEVGR	0x40000014	Event Generation Register	0x0000
R16_TIM2_CHCTLR1	0x40000018	Compare/Capture Control Register 1	0x0000
R16_TIM2_CHCTLR2	0x4000001C	Compare/Capture Control Register 2	0x0000
R16_TIM2_CCER	0x40000020	Compare/Capture Enable Register	0x0000
R16_TIM2_CNT	0x40000024	Counter of General-Purpose Timer	0x0000
R16_TIM2_PSC	0x40000028	Count Clock Prescaler	0x0000
R16_TIM2_ATRLR	0x4000002C	Auto-reload Register	0xFFFF
R32_TIM2_CH1CVR	0x40000034	Compare/Capture Register 1	0x00000000
R32_TIM2_CH2CVR	0x40000038	Compare/Capture Register 2	0x00000000
R32_TIM2_CH3CVR	0x4000003C	Compare/Capture Register 3	0x00000000
R32_TIM2_CH4CVR	0x40000040	Compare/Capture Register 4	0x00000000
R16_TIM2_DTCR	0x40000044	Dead-time Function Configuration Register	0x0000
R16_TIM2_DMACFGR	0x40000048	DMA Control Register	0x0000
R16_TIM2_DMAADR	0x4000004C	DMA Address Register in Continuous Mode	0x0000

12.4.1 Control Register 1 (TIM2_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPLVL	CAPOV	Reserved				CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN

Bit	Name	Access	Description	Reset value
15	CAPLVL	RW	In the double edge capture mode, the capture level is indicated to enable. 0: Disable indication function; 1: Enable indication function. *Note: After enabling, [16] of CHxCVR indicates the level corresponding to the capture value.*	0
14	CAPOV	RW	Capture value mode configuration. 0: The capture value is the value of the actual counter; 1: When a counter overflow occurs before capture, the CHxCVR value is 0xFFFF.	0

[13:10]	Reserved	RO	Reserved	0
[9:8]	CKD[1:0]	RW	These two bits define the frequency division ratio between the timer clock (CK_INT) frequency and the sampling clock used by the digital filter. 00: Tdts=Tck_int; 01: Tdts= 2xTck_int; 10: Tdts= 4xTck_int; 11: Reserved.	0
7	ARPE	RW	Automatically reload the pre-installed enable position. 1: Enable automatic reload value registers (ATRLR); 0: Disable automatic reload value registers (ATRLR).	0
[6:5]	CMS[1:0]	RW	Center alignment mode selection. . 00: Edge alignment mode. The counter counts up or down according to the direction bit (DIR); 01: Center alignment mode 1. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured for output (CCxS=00 in the CHCTLRx register) is set only when the counter counts down; 10: Center alignment mode 2. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured for output (CCxS=00 in the CHCTLRx register) is set only when the counter counts up; 11: Center alignment mode 3. The counter counts up and down alternately. The output comparison interrupt flag bit of the channel configured for output (CCxS=00 in the CHCTLRx register) is set when the counter counts up and down. *Note: Switching from edge alignment mode to center alignment mode is not allowed when the counter is enabled (CEN=1).*	0
4	DIR	RW	Counter direction. 1: The counting mode of the counter is subtractive counting 0: The counting mode of the counter is incrementing. *Note: This bit is not valid when the counter is configured in central alignment mode or encoder mode.*	0
3	OPM	RW	Mono-pulse mode. 1: The counter stops when the next update event (clearing the CEN bit) occurs. 0: The counter does not stop when the next update event occurs.	0
2	URS	RW	Update the request source, through which the software	0

			selects the source of the UEV event. 1: If an update interrupt or DMA request is enabled, an update interrupt or DMA request occurs only if the counter overflows / underflows 0: If an update interrupt or DMA request is enabled, any of the following events generates an update interrupt or DMA request: - Counter overflow / underflow - Set the UG bit -Updates generated by slave mode controller	
1	UDIS	RW	Disable updates, and the software enables / disables the generation of UEV events through this bit. 1: Disable UEV. No update events are generated, and the registers (ATRLR, PSC, CHCTLRx) hold their values. If the UG bit is set or a hardware reset is issued from the mode controller, the counter and prescaler are reinitialized. 0: Enable UEV. The UEV event is generated by any of the following events: -Counter overflow / underflow -Set UG bit - Updates generated by slave mode controller Registers with caches are loaded into their preloaded values.	0
0	CEN	RW	Counter enable 1: Enable counter; 0: Disable counter. *Note: The external clock, gating mode and encoder mode will not work until the CEN bit is set in the software. The trigger mode automatically sets the CEN bit through the hardware.*	0

12.4.2 Control Register 2 (TIM2_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TI1S	MMS[2:0]			CCDS	Reserved		

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TI1S	RW	TI1 selection: 1: TIM2_CH1, TIM2_CH2, and TIM2_CH3 pins are XOR connected to the TI1 input; 0: TIM2_CH1 pin is connected directly to the TI1 input.	0

[6:4]	MMS[2:0]	RW	Master mode selection: these 3 bits are used to select the synchronization information (TRGO) sent to the slave timer in master mode. Possible combinations are as follows: 000: The reset-UG bit is used as a trigger output (TRGO). If the reset is triggered by the input (the slave mode controller is in reset mode), the signal on the TRGO will have a delay relative to the actual reset; 001: The enable-counter enable signal CNT_EN is used as a trigger output (TRGO). Sometimes it is necessary to start multiple timers at the same time or control to enable slave timers over a period of time. The counter enable signal is the logic or generation of the trigger input signal through the CEN control bit and gating mode. When the counter enable signal is controlled by the trigger input, there is a delay on the TRGO unless the master / slave mode is selected (see description of the MSM bit in the TIM2_SMCFGR register) 010: The update event is selected as the trigger input (TRGO). For example, the clock of a master timer can be used as a prescaler for a slave timer. 011: Compare pulses, when a capture or a comparison is successful, when the CC1IF flag is to be set (even if it is already high), trigger the output to send a positive pulse (TRGO); 100: The OC1REF signal is used as a trigger output (TRGO); 101: The OC2REF signal is used as a trigger output (TRGO); 110: The OC3REF signal is used as a trigger output (TRGO); 111: The OC4REF signal is used as a trigger output (TRGO);	0
3	CCDS	RW	1: When an update event occurs, a DMA request for CHxCVR is sent; 0: When CHxCVR occurs, a DMA request for CHxCVR is generated.	0
[2:0]	Reserved	RO	Reserved	0

12.4.3 Slave Mode Control Register (TIM2_SMCFGR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Reserved	SMS[2:0]		

Bit	Name	Access	Description	Reset value
15	ETP	RO	ETR triggers polarity selection, which selects whether to enter the ETR directly or the inversion of the ETR. 1: The ETR will be inverted, low or effective at falling edge; 0: ETR, high level or rising edge is effective.	0
14	ECE	RW	External clock mode 2 enable selection. 1: Enable external clock mode 2; 0: Disable external clock mode 2. *Note 1: Slave mode can be used in conjunction with external clock mode 2: reset mode, gated mode and trigger mode; however, TRGI cannot be connected to ETRF (TS bit cannot be 111b).* *Note 2: When external clock mode 1 and external clock mode 2 are enabled at the same time, the input of the external clock is ETRF.*	0
[13:12]	ETPS[1:0]	RW	External trigger signal (ETRP) frequency division, the maximum frequency of this signal cannot exceed the frequency of 1 TIM2CLK 4, can be reduced through this domain. 00: Disable pre-division; 01: ETRP frequency divided by 2; 10: ETRP frequency divided by 4; 11: ETRP frequency divided by 8.	0
[11:8]	ETF[3:0]	RW	External trigger filter, in fact, the digital filter is an event counter, which uses a certain sampling frequency to record N events and produce an output jump. 0000: No filter, sample as Fdts; 0001: Sampling frequency Fsampling=Fck_int, N=2; 0010: Sampling frequency Fsampling=Fck_int, N=4; 0011: Sampling frequency Fsampling=Fck_int, N=8; 0100: Sampling frequency Fsampling=Fdts/2, N=6; 0101: Sampling frequency Fsampling=Fdts/2, N=8; 0110: Sampling frequency Fsampling=Fdts/4, N=6; 0111: Sampling frequency Fsampling=Fdts/4, N=8; 1000: Sampling frequency Fsampling=Fdts/8, N=6; 1001: Sampling frequency Fsampling=Fdts/8, N=8; 1010: Sampling frequency Fsampling=Fdts/16, N=5; 1011: Sampling frequency Fsampling=Fdts/16, N=6; 1100: Sampling frequency Fsampling=Fdts/16, N=8; 1101: Sampling frequency Fsampling=Fdts/32, N=5; 1110: Sampling frequency Fsampling=Fdts/32, N=6; 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0
7	MSM	RW	Master / slave mode selection.	0

			1: The event on the trigger input (TRGI) is delayed to allow perfect synchronization between the current timer (via TRGO) and its slave timer. This is useful when requiring several timers to be synchronized to a single external event. 0: Does not function.	
[6:4]	TS[2:0]	RW	Trigger the selection field, and these three bits select the trigger input source used to synchronize the counter. 000: Internal trigger 0 (ITR0); 001: Internal trigger 1 (ITR1); 010: Internal trigger 2 (ITR2); 011: Internal trigger 3 (ITR3); 100: Edge detector of TI1 (TI1F_ED); 101: Filtered timer input 1 (TI1FP1); 110: Filtered timer input 2 (TI2FP2); 111: External trigger input (ETRF); The above can only be changed when SMS is 0.	0
3	Reserved	RO	Reserved	0
[2:0]	SMS[2:0]	RW	Enter the mode selection field. Select the clock and trigger mode of the core counter. 000: Driven by internal clock CK_INT; 001: Encoder mode 1, according to the level of the TI1FP1, the core counter increases or decreases at the edge of the TI2FP2; 010: Encoder mode 2, according to the level of the TI2FP2, the core counter increases or decreases the count at the edge of the TI1FP1; 011: Encoder mode 3, according to the input level of another signal, the core counter increases or decreases the count at the edge of the TI1FP1 and TI2FP2; 100: Reset mode, triggering the rising edge of the input (TRGI) initializes the counter and generates a signal to update the register 101: In gated mode, when the trigger input (TRGI) is high, the clock of the counter is turned on; when the trigger input becomes low, the counter stops, and the start and stop of the counter is controlled. 110: Trigger mode, the counter starts at the rising edge of the trigger input TRGI, and only the start of the counter is controlled; 111: External clock mode 1, the rising edge of the selected trigger input (TRGI) drives the counter.	0

12.4.4 TIM2 DMA/Interrupt Enable Register (TIM2_DMAINTENR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TDE	Reserved	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Reserved	TIE	Reserved	CC4IE	CC3IE	CC2IE	CC1IE	UIE

Bit	Name	Access	Description	Reset value
15	Reserved	RO	Reserved	0
14	TDE	RW	Trigger DMA request enable bit. 1: Enable trigger DMA request; 0: Disable trigger DMA request.	0
13	Reserved	RO	Reserved	0
12	CC4DE	RW	DMA request enable bit of compare/capture channel 4 1: Enable DMA request of compare/capture channel 4; 0: Disable DMA request of compare/capture channel 4.	0
11	CC3DE	RW	DMA request enable bit of compare/capture channel 3 1: Enable DMA request of compare/capture channel 3; 0: Disable DMA request of compare/capture channel 3.	0
10	CC2DE	RW	DMA request enable bit of compare/capture channel 2 1: Enable DMA request of compare/capture channel 2; 0: Disable DMA request of compare/capture channel 2.	0
9	CC1DE	RW	DMA request enable bit of compare/capture channel 1 1: Enable DMA request of compare/capture channel 1; 0: Disable DMA request of compare/capture channel 1.	0
8	UDE	RW	Updated DMA request enable bit. 1: Enable updated DMA request; 0: Disable updated DMA request.	0
7	Reserved	RO	Reserved	0
6	TIE	RW	Trigger interrupt enable bit. 1: Enable trigger interrupt; 0: Disable trigger interrupt.	0
5	Reserved	RO	Reserved	0
4	CC4IE	RW	Compare/capture channel 4 interrupt enable bit. 1: Enable compare/capture channel 4 interrupt; 0: Disable compare/capture channel 4 interrupt.	0
3	CC3IE	RW	Compare/capture channel 3 interrupt enable bit. 1: Enable compare/capture channel 3 interrupt; 0: Disable compare/capture channel 3 interrupt.	0
2	CC2IE	RW	Compare/capture channel 2 interrupt enable bit. 1: Enable compare/capture channel 2 interrupt; 0: Disable compare/capture channel 2 interrupt.	0
1	CC1IE	RW	Compare/capture channel 1 interrupt enable bit. 1: Enable compare/capture channel 1 interrupt; 0: Disable compare/capture channel 1 interrupt.	0

0	UIE	RW	Updated interrupt enable bit. 1: Enable updated interrupt; 0: Disable updated interrupt.	0

12.4.5 Interrupt Status Register (TIM2_INTFR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4OF	CC3OF	CC2OF	CC1OF	Reserved		TIF	Reserved	CC4IF	CC3IF	CC2IF	CC1IF	UIF

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4OF	RW0	Compare/capture channel 4 repeated capture flag bit.	0
11	CC3OF	RW0	Compare/capture channel 3 repeated capture flag bit.	0
10	CC2OF	RW0	Compare/capture channel 2 repeated capture flag bit.	0
9	CC1OF	RW0	Compare/capture channel 1 repeated capture flag bit, only when the capture channel is configured to input capture mode. The flag is set by hardware and can be cleared by software write 0. 1: When the value of the counter is captured to the capture compare register, the state of the CC1IF has been set 0: There is no repetitive capture.	0
[8:7]	Reserved	RO	Reserved	0
6	TIF	RW0	Trigger interrupt flag bit, when a trigger event occurs, the position bit is matched by the hardware and cleared by the software. The trigger event includes the detection of a valid edge at the TRGI input, or any edge in the gated mode, from a mode other than the gated mode. 1: Trigger event generation; 0: No trigger event generation.	0
5	Reserved	RO	Reserved	0
4	CC4IF	RW0	Compare/capture channel 4 interrupt flag bit.	0
3	CC3IF	RW0	Compare/capture channel 3 interrupt flag bit.	0
2	CC2IF	RW0	Compare/capture channel 2 interrupt flag bit.	0
1	CC1IF	RW0	Compare/capture channel 1 interrupt flag bit. If the compare/capture channel is configured as output mode: This bit is set by hardware when the counter value matches the comparison value, except in centrosymmetric mode. The bit is cleared by the software. 1: The value of the core counter matches the value of	0

			compare/capture register 1; 0: No matching occurs. If compare/capture channel 1 is configured as input mode When the capture event occurs, the bit is set by the hardware, which is cleared by the software or by reading compare/capture register. 1: Counter is captured compare/capture register; 0: No input capture generation.	
0	UIF	RW0	Updated interrupt flag bit. When updated event occurs, the bit is set by hardware, cleared by software. 1: Updated event generation; 0: No updated event generation. The following feature will generate updated event: If UDIS=0, when the repeat counter value overflows or underflows; If URS=0, UDIS=0, when the UG bit is set, or when the counter core counter is reinitialized by software; If URS=0, UDIS=0, when the counter CNT is reinitialized by the triggered event.	0

12.4.6 Event Generation Register (TIM2_SWEVGR)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved									TG	Reserved	CC4G	CC3G	CC2G	CC1G	UG

Bit	Name	Access	Description	Reset value
[15:7]	Reserved	RO	Reserved	0
6	TG	WO	Trigger event generation bit. This bit is set by software and zeroed by hardware to generate a trigger event. 1: A trigger event is generated, and the TIF is set. If the corresponding interrupt and DMA are enabled, the corresponding interrupt and DMA are generated. 0: No action.	0
5	Reserved	RO	Reserved	0
4	CC4G	WO	Compare/capture event generation 4, generates compare/capture event 4.	0
3	CC3G	WO	Compare/capture event generation 3, generates compare/capture event 3.	0
2	CC2G	WO	Compare/capture event generation 2, generates compare/capture event 2.	0
1	CC1G	WO	Compare/capture event generation 1, generates compare/capture event 1.	G

			This bit is set by software and zeroed by hardware. Used to generate a comparison capture event. 1: Generates a compare/capture event in compare/capture channel 1: If compare/capture channel 1 is configured as output: Set CC1IF bit. If enable corresponding interrupt and DMA, it will generate corresponding interrupt and DMA; If compare/capture channel 1 is configured as input: The current core counter' value is captured to compare/capture register 1; Set CC1IF bit, if enable corresponding interrupt and DMA, it will generate corresponding interrupt and DMA. If CC1IF is already set, set CC1OF bit. 0: No action.	
0	UG	WO	Updated event generation bit, generates event. The bit is set by software, cleared by hardware. 1: Initialize the counter and generate an update event; 0: No action. *Note: The counter of the prescaler is also cleared, but the prescaler coefficient remains the same. If the core counter is cleared in centrosymmetric mode or increment mode, the core counter takes the value of the value register if it is in subtractive mode.*	0

12.4.7 Compare/Capture Control Register 1 (TIM2_CHCTLR1)

Offset address: 0x18

The channel can be used for input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bits. The functions of other bits of the register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			

Compare mode (Pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC2CE	RW	Compare/capture channel 2 clear enable bit. 1: Once the ETRF input high level is detected, clear the OC2REF bit zero. 0: OC2REF is not affected by ETRF input.	0
[14:12]	OC2M[2:0]	RW	Compare capture channel 2 mode settings fields. The three bits define the action of outputting the	0

			reference signal OC2REF, while OC2REF determines the values of OC2 and OC2N. OC2REF is efficient at high levels, while the effective levels for OC2 and OC2N depend on the CC2P and CC2NP bits. 000: Frozen. The comparison between the value of the compare/capture register and the core counter has no effect on OC2REF; 001: Force to be set to a valid level. Force OC2REF to be high when the core counter is the same as the value of compare/capture register 2; 010: Force to be set to an invalid level. Force OC2REF to be low when the value of the core counter is the same as that of compare/capture register 2 011: Flip. Flip the level of the OC2REF when the core counter is the same as the value of the compare/ capture register 1; 100: Force to be set to an invalid level. Force OC2REF to be low; 101: Force to be set to a valid level. Force OC2REF to be high; 110: PWM Mode 1: In up count, once the core counter is less than the value of the compare capture register, channel 2 is valid level, otherwise it is invalid level; in downward counting, once the core counter is greater than the value of the compare capture register, channel 2 is invalid level (OC2REF=0), otherwise it is valid level (OC2REF=1); 111: PWM mode 2: In up count, once the core counter is smaller than the value of the compare capture register, channel 2 is invalid level, otherwise it is valid level; in down count, once the core counter is larger than the value of the compare capture register, channel 2 is valid level (OC2REF=1), otherwise it is invalid level (OC2REF=0). *Note: Once the LOCK level is set to 3 and CC2S=00b, this bit cannot be modified. In PWM mode 1 or PWM mode 2, the OC2REF level changes only when the comparison result changes or when the output compare mode is switched from frozen mode to PWM mode.*	
11	OC2PE	RW	Compare capture register 2 preload enable bit. 1: Enable preload function of the compare/capture register 2. The read and write operation only operates on the preload register. The preload value of the compare/capture register 2 is loaded into the current shadow register when the update event arrives.	0

			0: Disable preload function of the compare/capture register 2. Compare/capture register 2 can be written at any time and the newly written value takes effect immediately. *Note: Once the LOCK level is set to 3 and CC2S=00, this bit cannot be modified; only in mono-pulse mode (OPM=1) you can use PWM mode without confirming the preload register, otherwise its action is uncertain.*	
10	OC2FE	RW	Compare/capture channel 2 fast enable level. This bit is used to speed up the compare/capture channel of the response of output to triggered input events. 1: The input to the valid edge of the trigger acts as if a comparison match had occurred. Therefore, the OC is set to the comparison level independent of the comparison result. The delay between the effective edge of the sampling trigger and the output of the compare/capture channel 2 is shortened to three clock cycles; 0: According to the value of the counter and the compare capture register 1, compare/capture channel 2 operate normally, even if the trigger is turned on. When the input of the flip-flop has a valid edge, the minimum delay of activating the output of the compare/capture channel 2 is 5 clock cycles. *Note: OC2FE works only if the channel is configured in PWM1 or PWM2 mode.*	0
[9:8]	CC2S[1:0]	RW	Compare/capture channel 2 input selection fields. 00: Compare/capture channel 2 is configured to output; 01: Compare/capture channel 2 is configured as input and IC2 is mapped on TI2; 10: Compare/capture channel 2 is configured as input and IC2 is mapped on TI1; 11: Compare/capture channel 2 is configured as input and IC2 is mapped on TRC. This mode works only when the internal trigger input is selected (selected by the TS bit). *Note: Compare/capture channel 2 is writable only when the channel is closed (CC2E is 00:00).*	0
7	OC1CE	RW	Compare/capture channel 1 clear enable bit.	0
[6:4]	OC1M[2:0]	RW	Compare/capture channel 1 mode setting fields.	0
3	OC1PE	RW	Compare/capture register 1 reload enable bit.	0
2	OC1FE	RW	Compare/capture channel 1 fast enable bit.	0
[1:0]	CC1S[1:0]	RW	Compare/capture channel 1 input selection fields.	0

Capture mode (Pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC2F[3:0]	RW	Input capture filter 2 configuration domain. These bits set the sampling frequency of TI1 input and the length of digital filter. The digital filter consists of an event counter, which records N events and produces an output jump. 0000: No filter, sample as fDTS; 1000: Sampling frequency Fsampling=Fdts/8, N=6; 0001: Sampling frequency Fsampling=Fck_int, N=2; 1001: Sampling frequency Fsampling=Fdts/8, N=8; 0010: Sampling frequency Fsampling=Fck_int, N=4; 1010: Sampling frequency Fsampling=Fdts/16, N=5; 0011: Sampling frequency Fsampling=f=Fck_int, N=8; 1011: Sampling frequency Fsampling=Fdts/16, N=6; 0100: Sampling frequency Fsampling=Fdts/2, N=6; 1100: Sampling frequency Fsampling=Fdts/16, N=8; 0101: Sampling frequency Fsampling=Fdts/2, N=8; 1101: Sampling frequency Fsampling=Fdts/32, N=5; 0110: Sampling frequency Fsampling=Fdts/4, N=6; 1110: Sampling frequency Fsampling=Fdts/32, N=6; 0111: Sampling frequency Fsampling=Fdts/4, N=8; 1111: Sampling frequency Fsampling=Fdts/32, N=8.	0
[11:10]	IC2PSC[1:0]	RW	Compare/capture channel 2 prescaler configuration domain. These two bits define the pre-division coefficient of the comparison capture channel 2. Once CC1E=0, the prescaler resets. 00: Without a prescaler, each edge detected on the capture input triggers a capture.; 01: Capture is triggered every 2 events; 10: Capture is triggered every 4 events; 11: : Capture is triggered every 8 events.	0
[9:8]	CC2S[1:0]	RW	Compare/capture channel 2 input selection domain, these two bits define the direction of the channel (input / output), and the selection of the input pin. 00: Compare/capture channel 1 channel is configured to output; 01: Compare/capture channel 1 channel is configured as input and IC1 is mapped on TI1; 10: Compare/capture channel 1 channel is configured as input and IC1 is mapped on TI2; 11: Compare/capture channel 1 channel is configured as input and the IC1 is mapped on the TRC. This mode works only when the internal trigger input is selected (selected by the TS bit). *Note: CC1S is writable only when the channel is closed*	0

			(CC1E is 0).	
[7:4]	IC1F[3:0]	RW	Input capture filter 1 configuration domain.	0
[3:2]	IC1PSC[1:0]	RW	Compare/capture channel 1 prescaler configuration domain.	0
[1:0]	CC1S[1:0]	RW	Compare/capture channel 1 input selection domain.	0

12.4.8 Compare/Capture Control Register 2 (TIM2_CHCTLR2)

Offset address: 0x1C

The channel can be used for input (Capture mode) or output (Compare mode), and the direction of the channel is defined by the corresponding CCxS bits. The functions of other bits of the register are different in input and output modes. OCxx describes the function of the channel in output mode, and ICxx describes the function of the channel in input mode.

15	14 13 12	11	10	9 8	7	6 5 4	3	2	1 0
OC4CE	OC4M[2:0]	OC4PE	OC4FE	CC4S[1:0]	OC3CE	OC3M[2:0]	OC3PE	OC3FE	CC3S[1:0]
IC4F[3:0]		IC4PSC[1:0]			IC3F[3:0]		IC3PSC[1:0]		

Compare mode (pin direction is output).

Bit	Name	Access	Description	Reset value
15	OC4CE	RW	Compare/capture channel 4 clear enable bit.	0
[14:12]	OC4M[2:0]	RW	Compare/capture channel 4 mode setting domain.	0
11	OC4PE	RW	Compare/capture register 4 preload enable bit.	0
10	OC4FE	RW	Compare/capture channel 4 fast enable bit.	0
[9:8]	CC4S[1:0]	RW	Compare/capture channel 4 input selection domain.	0
7	OC3CE	RW	Compare/capture channel 3 clear enable bit.	0
[6:4]	OC3M[2:0]	RW	Compare/capture channel 3 mode setting domain.	0
3	OC3PE	RW	Compare/capture register 3 preload enable bit.	0
2	OC3FE	RW	Compare/capture channel 3 fast enable bit.	0
[1:0]	CC3S[1:0]	RW	Compare/capture channel 3 input selection domain.	0

Capture mode (pin direction is input).

Bit	Name	Access	Description	Reset value
[15:12]	IC4F[3:0]	RW	Input capture filter 4 configuration field.	0
[11:10]	IC4PSC[1:0]	RW	Compare/capture channel 4 prescaler configuration field.	0
[9:8]	CC4S[1:0]	RW	Compare/capture channel 4 input selection domain.	0
[7:4]	IC3F[3:0]	RW	Input capture filter 3 configuration field.	0
[3:2]	IC3PSC[1:0]	RW	Compare/capture channel 3 prescaler configuration field.	0
[1:0]	CC3S[1:0]	RW	Compare/capture channel 3 input selection domain.	0

12.4.9 Compare/Capture Enable Register (TIM2_CCER)

Offset address: 0x20

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CC4P	CC4E	Reserved		CC3P	CC3E	Reserved		CC2P	CC2E	Reserved		CC1P	CC1E

Bit	Name	Access	Description	Reset value
[15:14]	Reserved	RO	Reserved	0
13	CC4P	RW	Compare/capture channel 4 output polarity setting bit.	0
12	CC4E	RW	Compare/capture channel 4 output enable bit.	0
[11:10]	Reserved	RO	Reserved	0
9	CC3P	RW	Compare/capture channel 3 output polarity setting bit.	0
8	CC3E	RW	Compare/capture channel 3 output enable bit.	0
[7:6]	Reserved	RO	Reserved	0
5	CC2P	RW	Compare/capture channel 2 output polarity setting bit.	0
4	CC2E	RW	Compare/capture channel 2 output enable bit.	0
[3:2]	Reserved	RO	Reserved	0
1	CC1P	RW	Compare/capture channel 1 output polarity setting bit. CC1 channel configured as output: 1: OC1 active low. 0: OC1 active high. CC1 channel configured as input: The bit selects whether the inverse signal of IC1 or IC1 is used as the trigger or capture signal. 1: Invert: capture occurs on the falling edge of IC1; when used as an external trigger, IC1 inverts 0: No inversion: capture occurs on the rising edge of the IC1; when used as an external trigger, the IC1 is not inverted.	0
0	CC1E	RW	Compare capture channel 1 output enable bit. CC1 channel configured as output: 1: On: The OC1 signal is output to the corresponding output pin; 0: Off: OC1 disables output. CC1 channel configured as input: This bit determines whether the counter value can be captured into the TIM2_CH1CVR register. 1: Capture enable. 0: Capture disable.	0

12.4.10 Counter for General-purpose Timer (TIM2_CNT)

Offset address: 0x24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CNT[15:0]

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the timer's counter.	0

12.4.11 Counting Clock Prescaler (TIM2_PSC)

Offset address: 0x28

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSC[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	PSC[15:0]	RW	The dividing factor of the prescaler of the timer; the clock frequency of the counter is equal to the input frequency of the divider/(PSC+1).	0

12.4.12 Auto-reload Value Register (TIM2_ATRLR)

Offset address: 0x28

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATRLR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	ATRLR[15:0]	RW	The value of ATRLR[15:0] will be loaded into the counter, read section 12.2.4 for when ATRLR acts and updates; the counter stops when ATRLR is empty.	0xFFFF

12.4.13 Compare/Capture Register 1 (TIM2_CH1CVR)

Offset address: 0x34

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL1	RO	The level indication bit corresponding to the captured value	0
[15:0]	CH1CVR[15:0]	RW	The value compare/capture register channel 1.	0

12.4.14 Compare/Capture Register 2 (TIM2_CH2CVR)

Offset address: 0x38

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL2
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL2	RO	The level indication bit corresponding to the captured value	0
[15:0]	CH2CVR[15:0]	RW	The value compare/capture register channel 2.	0

12.4.15 Compare/Capture Register 3 (TIM2_CH3CVR)

Offset address: 0x3C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL3
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL3	RO	The level indication bit corresponding to the captured value	0
[15:0]	CH3CVR[15:0]	RW	The value compare/capture register channel 3.	0

12.4.16 Compare/Capture Register 4 (TIM2_CH4CVR)

Offset address: 0x40

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															LEVEL4
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR[15：0]															

Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	LEVEL4	RO	The level indication bit corresponding to the captured value	0
[15:0]	CH4CVR[15:0]	RW	The value compare/capture register channel 4.	0

12.4.17 Dead-time Function Configuration Register (TIM2_DTCR)

Offset address: 0x44

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DT2				DT1				Reserved		DT2N_P	DT2_P	DT1N_P	DT1_P	OC2N_EN	OC1N_EN

Bit	Name	Access	Description	Reset value
[15:12]	DT2	RW	Channel 2 dead-time configuration: T is the module clock of TIM2. 0000: Dead-time is 1*T; 0001: Dead-time is 2*T; … 1110: Dead-time is 15*T; 1111: Dead-time is 16*T. *Note: The width of the dead-time must be less than the effective level width.*	0
[11:8]	DT1	RW	Channel 2 dead-time configuration: T is the module clock of TIM2. 0000: Dead-time is 1*T; 0001: Dead-time is 2*T; … 1110: Dead-time is 15*T; 1111: Dead-time is 16*T. *Note: The width of the dead-time must be less than the effective level width.*	0
[7:6]	Reserved	RO	Reserved	0
5	DT2N_P	RW	Complementary output function-Channel 2 complementary channel output polarity setting bit: 1: Active low; 0: Active high.	0
4	DT2_P	RW	Complementary output function-Channel 2 output polarity setting bit: 1: Active low; 0: Active high.	0
3	DT1N_P	RW	Complementary output function-Channel 1 complementary channel output polarity setting bit: 1: Active low; 0: Active high.	0
2	DT1_P	RW	Complementary output function-Channel 1 output polarity setting bit: 1: Active low; 0: Active high.	0

1	OC2N_EN	RW	Complementary output function – Channel 2 and complementary channel output enable bit (Channel 4 is the complementary channel of channel 2): 1: Dead-time function of channel 2 enables, complementary channel alternate channel 4; 0: Dead-time function of channel 2 disables.	0
0	OC1N_EN	RW	Complementary output function – Channel 1 and complementary channel output enable bit (Channel 3 is the complementary channel of channel 1): 1: Dead-time function of channel 1 enables, complementary channel alternate channel 3; 0: Dead-time function of channel 1 disables.	0

12.4.18 DMA Control Register (TIM2_DMACFGR)

Offset address: 0x48

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			DBL[4:0]					Reserved			DBA[4:0]				

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
[12:8]	DBL[4:0]	RW	The length of the DMA continuous transmission, the actual value of which is the value of this field + 1.	0
[7:5]	Reserved	RO	Reserved	0
[4:0]	DBA[4:0]	RW	These bits define the offset of the DMA in continuous mode from the address where control register 1 is located.	0

12.4.18 DMA Address Register in Continuous Mode (TIM2_DMAADR)

Offset address: 0x4C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	DMAADR[15:0]	RW	The DMA address in continuous mode.	0

Chapter 13 Streamlined Timer (SLTM)

The module description in this chapter applies only to CH32V006, CH32V007, and CH32M007 products.

The Streamlined Timer module contains a 16-bit timer TIM3 that can be reinstalled automatically, which is used to generate pulses of a specific frequency (non-output) with TIM1 and ADC modules.

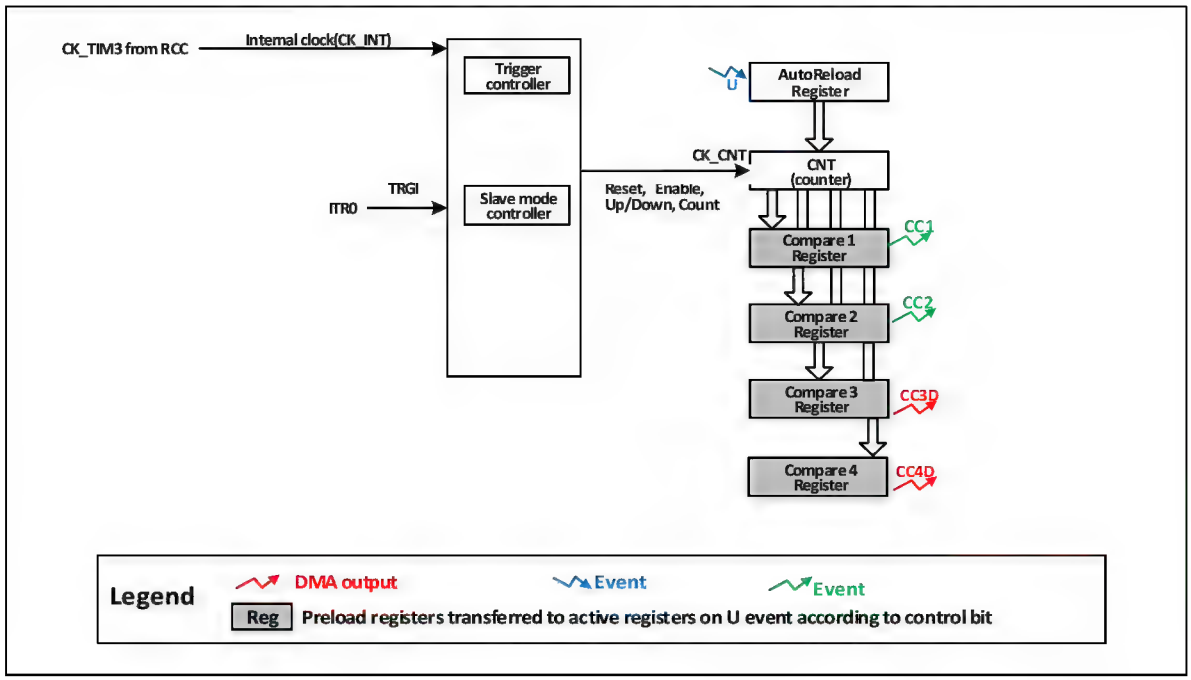
13.1 Main Features

The main features of the streamlined timer include:

- 16-bit auto-reload counter, supports incremental counting mode, decremental counting mode, incremental and decremental counting mode
- Support 4 independent compare channels, only support output comparison
- Support cascade synchronization with TIM1.
- Support for using DMA

13.2 Principle and Structure

Figure 13-1 Block diagram of the structure of the streamlined timer



13.2.1 Overview

As shown in Figure 13-1, the structure of the streamlined timer can be roughly divided into three parts, namely the input clock part, the core counter part and the compare channel part.

The clock of the streamlined timer can come from the HB bus clock (CK_INT) and from other timers (ITR1) with clock output function. After the set filtering operation, these input clock signals become CK_PSC clocks and output to the core counter.

The core of the streamlined timer is a 16-bit counter (CNT). CK_PSC input directly to CNT, CNT without frequency division supports incremental counting mode, decremental counting mode and incremental or decremental counting

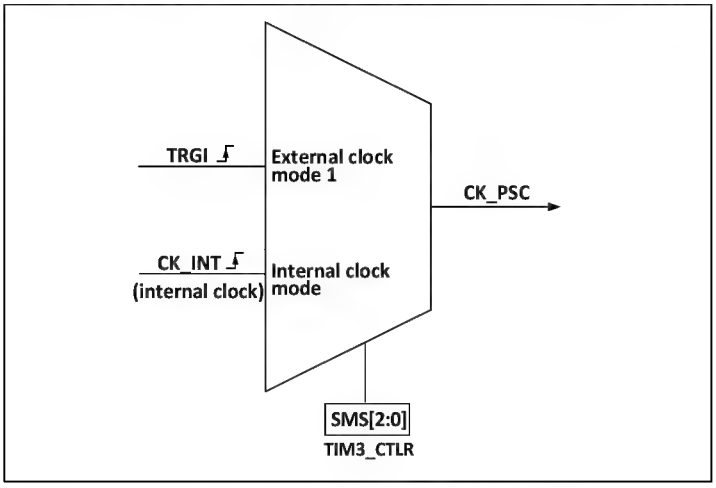
mode, and has an automatic reload value register (ATRLR) to reload initialization values for CNT at the end of each counting cycle.

The streamlined timer has four sets of compare channels, each of which has a set of compare registers (CHxCVR). Channels 1 and 2 support comparison with the master counter (CNT) and output pulses, that is, only the output mode; channels 3 and 4 support the comparison of the main counter (CNT) to generate DMA requests.

13.2.2 Clock Input

This section discusses the source of CK_PSC. The clock source part of the overall structure block diagram of the streamlined timer is intercepted here.

Figure 13-2 Streamlined timer CK_PSC source block diagram



The optional input clocks can be divided into 2 categories.

1) Internal HB clock input route: CK_INT
2) Input from other internal timers: ITRx.

By determining the input pulse selection of the SMS from which the CK_PSC comes from, the actual operation can be divided into two categories:

1) Select internal clock source (CK_INT)
2) External clock source mode 1

The two clock sources mentioned above can be selected by these two operations.

13.2.2.1 Internal Clock Source (CK_INT)

If the compact timer is started when the SMS domain is kept at 000b, the internal clock source (CK_INT) is selected as the clock. At this point, CK_INT is CK_PSC.

13.2.2.2 External Clock Source Mode 1

If the SMS domain is set to 111b, external clock source mode 1 is enabled. When the external clock source 1 is enabled, the source from which TRGI is selected as the CK_PSC is the internally triggered ITR1.

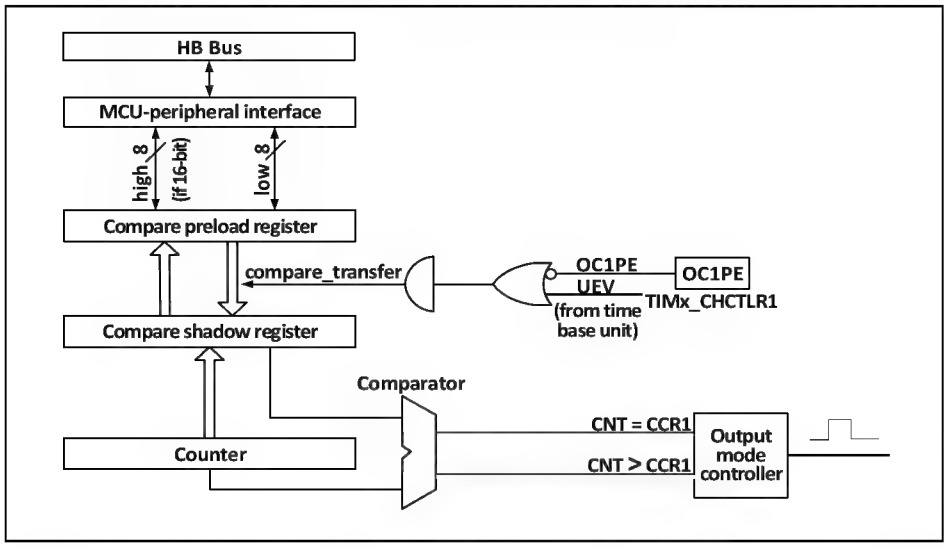
13.2.3 Counters and Peripherals

CK_PSC is directly CK_INT without frequency division. Reduced timer no counting clock pre-division function.

13.2.4 Compare Channel

The compare channel is the core of the timer to achieve complex functions, and its core is the compare register, supplemented by the output comparator. The structural block diagram of the compare channel is shown in figure 13-3.

Figure 13-3 Block diagram of the structure of the compare channel



The compare register consists of a preload register and a shadow register, and the read and write process operates only the preload register. During comparison, the contents of the preloaded register are copied to the shadow register, and then the contents of the shadow register are compared with the core counter (CNT).

13.3 Function and Implementation

The complex functions of the streamlined timer are realized by operating the compare channel of the timer, the clock input circuit, the counter and the surrounding components. The clock input of the timer may come from two clock sources including an internal clock source and trigger inputs of other timers. The operation of comparing the register channel and the clock source directly determines its function. The compare channel is unidirectional and only works in output mode.

13.3.1 Compare Output Mode

Compare output mode is one of the basic functions of timer. The principle of the compare output mode is that when the value of the core counter (CNT) is consistent with the value of the compare register, channel 1 ~ 2 outputs a specific waveform. Channel 3pr 4 generates a DMA request if the CCxDE bit is set in advance when a relatively consistent event is generated.

The steps to configure to compare output modes are as follows:

1) Configure the clock source and automatic reload values of the core counter (CNT)
2) Set the count value to be compared to the compare register (R16_TIMx_CHxCVR)
3) Keep OCxPE to 0 and disable preloaded registers that compare the obtained registers
4) Set the CEN bit to start the timer.

13.3.2 PWM Output Mode

PWM output mode is one of the basic functions of timer. The most common method of PWM output mode is to determine the PWM frequency using reinstalled values and to determine the duty cycle using comparison registers.

Set the OCxPE bit to enable the preloading register, and finally set the ARPE bit to enable the automatic reloading of the preloading register. When an update event occurs, the value of the preloaded register can be sent to the shadow register. In PWM mode, the core counter and the compare register are compared all the time, and according to the CMS bit, the timer can output edge-aligned or center-aligned PWM signals.

1) Edge alignment

When edge alignment is used, the core counter increases or decreases, OCxREF rises to high when the value of the core counter is greater than the compare register, and OCxREF decreases to low when the value of the core counter is smaller than the compare register.

2) Central alignment

When using the central alignment modes, the core counter runs in alternating incremental and decremental count modes, and OCxREF performs rising and falling jumps when the values of the core counter and the compare register match. However, the comparison flags are set at different times in the three central alignment modes.

13.3.3 Timer Synchronization Mode

The timer can only receive input from timer 1 (ITRx). The connection is triggered inside the timer as shown in Table 13-1.

Table 13-1 GTPM internal trigger connection

Slave timer	ITR0 (TS=000)	ITR1 (TS=001)	ITR2 (TS=010)	ITR3 (TS=011)
TIM3	TIM1	-	-	-

13.3.4 Debug Mode

When the system enters the debug mode, the timer can be controlled to continue running or stop according to the setting of DBG module.

13.4 Register Description

Table 13-2 TIM3-related registers list

Name	Offset address	Description	Reset value
R16_TIM3_CTLR	0x40000800	Control Register	0x0000
R16_TIM3_DMAINTENR	0x40000804	DMA Enable Register	0x0000
R16_TIM3_CNT	0x40000808	Timer Counter	0x0000
R16_TIM3_ATRLR	0x4000080C	Auto-reload Register	0xFFFF
R16_TIM3_CH1CVR	0x40000810	Compare Register 1	0x0000
R16_TIM3_CH2CVR	0x40000814	Compare Register 2	0x0000
R16_TIM3_CH3CVR	0x40000818	Compare Register 3	0x0000
R16_TIM3_CH4CVR	0x4000081C	Compare Register 4	0x0000

13.4.1 Control Register 1 (TIM3_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					SMS			ARPE	CMS[1:0]		DIR	Reserved		UDIS	CEN

Bit	Name	Access	Description	Reset value
[15:11]	Reserved	RO	Reserved	0
[10:8]	SMS	RW	000: Driven by internal clock CK_INT; 001: Reserved; 010: Reserved; 011: Reserved; 100: Reset mode, triggering the rising edge of the input (TRGI) initializes the counter and generates a signal to update the register; 101: In gated mode, when the trigger input (TRGI) is high, the clock of the counter is turned on; when the trigger input becomes low, the counter stops, and the start and stop of the counter is controlled. 110: Trigger mode, the counter starts at the rising edge of the trigger input TRGI, and only the start of the counter is controlled 111: External clock mode 1, the rising edge of the selected trigger input (TRGI) drives the counter.	0
7	ARPE	RW	Auto-reload preload enable bit. 1: Enable auto-reload value register (ATRLR); 0: Disable auto-reload value register (ATRLR).	0
[6:5]	CMS[1:0]	RW	Count mode selection: 00: Edge alignment mode. The counter counts up or down according to the direction bit (DIR); 01: Center alignment mode 1. The counter counts up and down alternately. The output of the channel compares the DMA flag bit and is set only when the counter is counted down; 10: Center alignment mode 2. The counter counts up and down alternately. The output of the channel compares the DMA flag bit and is set only when the counter is counted up. 11: Center alignment mode 3. The counter counts up and down alternately. The output of the channel compares the DMA flag bit, which is set when the counter counts up and down. *Note: Mode switching is not allowed when counter is enabled (CEN=1).*	0
4	DIR	RW	Counter direction. 1: The counting mode of the counter is decremental counting; 0: The counting mode of the counter is incremental counting.	0

			Note: This bit is not valid when the counter is configured in central alignment mode.	
[3:2]	Reserved	RO	Reserved	0
1	UDIS	RW	Updates disabled, and the software allows / disables the generation of UEV events through this bit. 1: Disable UEV. No update events are generated, and the registers (ARR, CHxCVR) maintain their values. If a hardware reset is issued from the mode controller, the counter is reinitialized 0: Enable UEV. The UEV event is generated by any of the following events: - Counter overflow / underflow - Updates generated by slave mode controller Registers with caches are loaded into their preload values	0
0	CEN	RW	Counter enable 1: Enable counter; 0: Disable counter. *Note: The external clock and gating mode cannot work until CEN bit is set in the software. The trigger mode automatically sets the CEN bit through the hardware.*	0

13.4.2 DMA Enable Register (TIM3_DMAINTENR)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CC4DE	CC3DE	Reserved							OC4PE	OC3PE	OC2PE	OC1PE

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	CC4DE	RW	DMA request enable bit of compare channel 4; 1: Enable; 0: Disable.	0
11	CC3DE	RW	DMA request enable bit of compare channel 3; 1: Enable; 0: Disable.	0
[10:4]	Reserved	RO	Reserved	0
3	OC4PE	RW	Compare register 4 preload enable bits. 1: The preload function of compare register 4 is enabled. The read and write operation only operates on the preload register. The preload value of compare register 4 is loaded into the current shadow register when the update event arrives.	0

			0: The preloading function of the compare register 4 is disabled, the compare register 4 can be written at any time, and the newly written value takes effect immediately.	
2	OC3PE	RW	Compare register 3 preload enable bits. 1: The preload function of compare register 3 is enabled. The read and write operation only operates on the preload register. The preload value of compare register 3 is loaded into the current shadow register when the update event arrives. 0: The preloading function of the compare register 3 is disabled, the compare register 3 can be written at any time, and the newly written value takes effect immediately.	0
1	OC2PE	RW	Compare register 2 preload enable bits. 1: The preload function of compare register 2 is enabled. The read and write operation only operates on the preload register. The preload value of compare register 2 is loaded into the current shadow register when the update event arrives. 0: The preloading function of the compare register 2 is disabled, the compare register 2 can be written at any time, and the newly written value takes effect immediately.	0
0	OC1PE	RW	Compare register 1 preload enable bits. 1: The preload function of compare register 1 is enabled. The read and write operation only operates on the preload register. The preload value of compare register 1 is loaded into the current shadow register when the update event arrives. 0: The preloading function of the compare register 1 is disabled, the compare register 1 can be written at any time, and the newly written value takes effect immediately.	0

13.4.3 Counter of Timer (TIM3_CNT)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CNT[15:0]	RW	The real-time value of the counter for the timer.	0

13.4.4 Auto-reload Register (TIM3_ATRLR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ATRLR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	ATRLR[15:0]	RW	The value of ATRLR [15:0] will be loaded into the counter; when ATRLR is empty, the counter stops.	0xFFFF

13.4.5 Compare Register 1 (TIM3_CH1CVR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH1CVR[15:0]	RW	The value of compare register channel 1.	0

13.4.6 Compare Register 2 (TIM3_CH2CVR)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH2CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH2CVR[15:0]	RW	The value of compare register channel 2.	0

13.4.7 Compare Register 3 (TIM3_CH3CVR)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH3CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH3CVR[15:0]	RW	The value of compare register channel 3.	0

13.4.8 Compare Register 4 (TIM3_CH4CVR)

Offset address: 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH4CVR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	CH4CVR[15:0]	RW	The value of compare register channel 4.	0

Chapter 14 Universal Synchronous Asynchronous Receiver Transmitter (USART)

This chapter USART1 module description applies to the full range of CH32V00X products; this chapter USART2 module description applies only to CH32V005, CH32V006, CH32V007 and CH32M007 products.

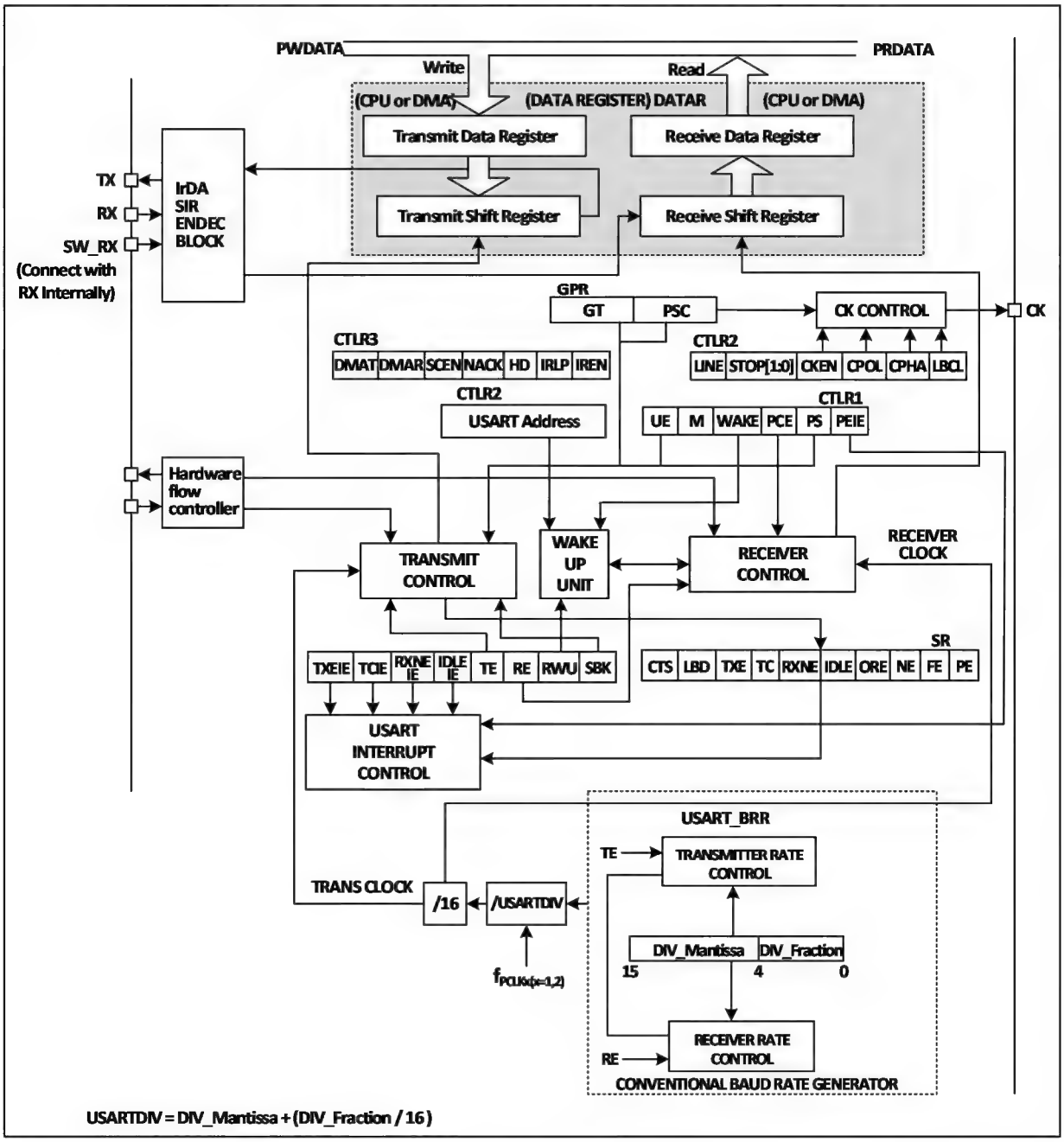
The module contains two general asynchronous transceivers USART1 and USART2.

14.1 Main Features

- Full-duplex or half-duplex synchronous or asynchronous communication
- NRZ data format
- Fractional baud rate generator, up to 3Mbps
- Programmable data length
- Configurable stop bits
- Support LIN, IrDA encoders, smart cards
- DMA support
- Multiple interrupt sources

14.2 Overview

Figure 14-1 Block diagram of synchronous/asynchronous transceiver



When the TE (transmit enable bit) is set, the data in the transmit shift register is output on the TX pin. When transmitting, the least significant bits are removed first, and each data frame starts with a low-level start bit, and then the transmitter sends eight-or nine-bit data words according to the setting on the M (word length) bit, and finally the number of configurable stop bits. If there is a parity bit, the last bit of the data word is the parity bit. After the TE is set, an idle frame is sent, which is 10-or 11-bit high, including the stop bit. The break frame is 10-or 11-bit low, followed by the stop bit.

14.3 Baud Rate Generator

The baud rate of the transceiver = HCLK/ (16*USARTDIV), and HCLK is the clock of HB. The value of USARTDIV is determined based on the fields DIV_M and DIV_F in USART_BRR. The formula for calculation is:

USARTDIV = DIV_M + (DIV_F/16)

It should be noted that the baud rate generated by the baud rate generator may not always generate just the baud rate required by the user, which may be biased. In addition to taking as close a value as possible, the way to reduce the deviation can also be to increase the HB clock. For example, when the baud rate is set to 76800bps, the USARTDIV value is set to 39.0625, and you can get exactly the baud rate of 76800bp at the highest frequency (48MHz), but if you need the baud rate of 921600bps, the calculated USARTDIV is 3.255, but in fact, the value filled in USART_BRR is only 3.25, the actual baud rate is 923076bps, with an error of 0.16%.

When the serial port waveform sent by the sender is transmitted to the receiver, the baud rate of the receiver and the sender has a certain error. The error mainly comes from three aspects: the actual baud rate of the receiver and the sender is different; the clock of the receiver and the sender has an error; the waveform changes in the line. The receiver of the peripheral module has a certain ability to receive tolerance. when the sum of the total deviation of the above three aspects is less than the tolerance limit of the module, the total deviation does not affect the transceiver. The tolerance limit of the module is affected by the use of fractional baud rate and M-bit (data field word length). The use of fractional baud rate and the use of 9-bit data field length will reduce the tolerance limit, but not less than 3%.

14.4 1-wire Half-duplex Mode

Half-duplex mode supports the use of a single pin (Using only TX pins) to receive and transmit, with TX pins and RX pins connected inside the chip.

The way to turn on half-duplex mode is to turn on the HDSEL position bit of control register 3 (R32_USARTx_CTLR3), but at the same time you need to turn off LIN mode and infrared mode, that is, to ensure that the IREN bit is reset, which is in control register 3 (R32_USARTx_CTLR3).

After setting it to half-duplex mode, you need to set the IO port of TX to multiplexing output high mode. In the case of TE setting, as long as the data is written to the data register, it will be sent. In particular, half-duplex mode may cause bus conflicts when multiple devices use single-bus transceiver, which needs to be avoided by users with software.

14.5 IrDA

The USART module supports the control of IrDA infrared transceivers for physical layer communication. To use IrDA, you must clear the LINEN, STOP, and HDSEL bits. NRZ (Non-return-to-zero) coding is used between the USART module and the SIR physical layer (Infrared transceiver) to support up to 115200bps rates.

IrDA is a half-duplex protocol. If USART is sending data to SIR, the IrDA decoder will ignore the new infrared signal. If USART is receiving data from SIR, SIR will not accept the signal from USART. The level logic sent by USART to SIR and SIR to USART is different. In the SIR receiving logic, the high level is 1 and the low level is 0, but in the SIR sending logic, the high level is 0 and the low level is 1.

14.6 DMA

The USART module supports the DMA function, which can be utilized to achieve fast continuous sending and

receiving. When DMA is enabled, the DMA writes data from a set memory space to the transmit buffer when TXE is set. When receiving with DMA, each time RXNE is set, DMA transfers the data in the receive buffer to a specific memory space.

14.7 Interrupts

The USART module supports a variety of interrupt sources, including transmit data register empty (TXE), CTS, transmit complete (TC), receive data ready (RXNE), data overflow (ORE), line idle (IDLE), parity error (PE), disconnect flag (LBD), noise (NE), overflow for multi-buffered communication (ORT), and frame error (FE), among others.

Table 14-1 Relationship between interrupts and corresponding enable bits

Interrupt source	Enable bit
Transmit data register empty (TXE)	TXEIE
Allowed to send (CTS)	CTSIE
Transmission complete (TC)	TCIE
Received data ready to be read (RXNE)	RXNEIE
Overrun error detected (ORE)	
Idle line detected (IDLE)	IDLEIE
Parity error (PE)	PEIE
Break flag (LBD)	LBDIE
Noise flag (NE)	EIE
Overflow of multi-buffered communication (ORE)	
Frame error for multibuffered communication (FE)	

14.8 Register Description

Table 14-2 USART1-related registers list

Name	Offset address	Description	Reset value
R32_USART1_STATR	0x40013800	UASRT Status Register	0x000000C0
R32_USART1_DATAR	0x40013804	UASRT Data Register	0x00000XXX
R32_USART1_BRR	0x40013808	UASRT Baud Rate Register	0x00000000
R32_USART1_CTLR1	0x4001380C	UASRT Control Register 1	0x00000000
R32_USART1_CTLR2	0x40013810	UASRT Control Register 2	0x00000000
R32_USART1_CTLR3	0x40013814	UASRT Control Register 3	0x00000000
R32_USART1_GPR	0x40013818	USART Prescaler Register	0x00000000

Table 14-3 USART2-related registers list

Name	Offset address	Description	Reset value
R32_USART2_STATR	0x40004400	UASRT Status Register	0x000000C0

R32_USART2_DATAR	0x40004404	UASRT Data Register	0x00000XXX
R32_USART2_BRR	0x40004408	UASRT Baud Rate Register	0x00000000
R32_USART2_CTLR1	0x4000440C	UASRT Control Register 1	0x00000000
R32_USART2_CTLR2	0x40004410	UASRT Control Register 2	0x00000000
R32_USART2_CTLR3	0x40004414	UASRT Control Register 3	0x00000000
R32_USART2_GPR	0x40004418	USART Prescaler Register	0x00000000

14.8.1 USART Status Register (USARTx_STATR) (x=1/2)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved						CTS	LBD	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE

Bit	Name	Access	Description	Reset value
[31:10]	Reserved	RO	Reserved	0
9	CTS	RW0	CTS status change flag. If the CTSE bit is set, when the nCTS output state changes, the bit will be set high by the hardware. Zero is cleared by the software. If the CTSIE bit has been set, an interrupt occurs. 1: There is a change in the nCTS state line; 0: There is no change on the nCTS state line.	0
8	LBD	RW0	LINBreak detection flag. When a LINBreak is detected, the bit is set by the hardware. Zero is cleared by the software. If the LBDIE is already set, an interrupt will occur. 1: LIN Break detected; 0: No pending LIN Break was detected.	0
7	TXE	RO	Transmit data register null flag. When the data in the TDR register is transferred by hardware to the shift register, the bit is set by the hardware. If the TXEIE has been set, an interrupt will occur, write to the data register, and this bit will be reset. 1: The data has been transferred to the shift register; 0: The data has not been transferred to the shift register.	1
6	TC	RW0	Transmit completion flag. When a frame containing data is sent and the TXE is set, the hardware will have this bit, and if the TCIE is set,	1

			there will be a corresponding interrupt, which will be cleared by the software after reading this bit and writing the data register. You can also write 0 directly to clear this bit. 1: Transmit completed; 0: The transmission is not completed yet.	
5	RXNE	RW0	Read the data register non-empty flag, when the data in the shift register is transferred to the data register, the bit will be set by the hardware. If the RXNEIE has already been set, a corresponding interrupt will also occur. The bit can be cleared by reading the data register. You can also write 0 directly to clear the bit. 1: The data is received and can be read; 0: The data hasn't been received yet.	0
4	IDLE	RO	Bus idle flag. When the bus is idle, the bit will be set by the hardware. If the IDLEIE is already set, a corresponding interrupt occurs. Reading the status register and then reading the data register clears this bit. 1: The bus is idle; 0: No idle bus was detected. *Note: This bit will not be set again until RXNE is set.*	0
3	ORE	RO	Overload error flag. This bit will be set when there is data in the receiving shift register that needs to be transferred to the data register, but there is still data in the receiving domain of the data register that has not been read. If RXNEIE is set, there will also be a corresponding interrupt. 1: Overload error occurred; 0: There is no overload error. *Note: When an overload error occurs, the value of the data register will not be lost, but the value of the shift register will be overwritten. If the EIE bit is set, the ORE flag position bit will be interrupted in multi-buffer communication mode.*	0
2	NE	RO	Noise error flag. When a noise error flag is detected, it is set by the hardware. After reading the status register, the operation of reading the data register resets this bit. 1: Noise detected; 0: No noise was detected. *Note: There is no interruption in this bit. If the EIE*	0

			bit is set, the FE flag position bit will be interrupted in multi-buffer communication mode.	
1	FE	RO	Frame error flag. When synchronization errors, excessive noise or breakers are detected, the bit will be set by the hardware. The operation of reading this bit and then reading the data register resets this bit. 1: Frame error detected; 0: No frame error was detected. *Note: This bit will not cause interruption. If the EIE bit is set, the FE flag position bit will be interrupted in multi-buffer communication mode.*	0
0	PE	RO	Check error flag. In the receive mode, if a parity error occurs, the hardware sets this bit. The operation of reading this bit and then reading the data register resets this bit. The software must wait for the RXNE flag bit to be set before clearing this bit. If the PEIE has been set before, then setting this bit will result in a corresponding interrupt. 1: A parity error occurred; 0: There are no check errors.	0

14.8.2 USART Data Register (USARTx_DATAR) (x=1/2)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							DR[8:0]								

Bit	Name	Access	Description	Reset value
[31:9]	Reserved	RO	Reserved	0
[8:0]	DR[8:0]	RW	Data register. This register is actually composed of two registers: the receive data register (RDR) and the transmit data register (TDR). The read and write operations of DR start with the read receive register (RDR) and the write send register (TDR).	X

14.8.3 USART Baud Rate Register (USARTx_BRR) (x=1/2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

DIV_Mantissa[11:0]	DIV_Fraction[3:0]

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
[15:4]	DIV_Mantissa[11:0]	RW	These 12 bits define the integer portion of the divider division factor.	0
[3:0]	DIV_Fraction[3:0]	RW	These four digits define the decimal part of the divider division factor.	0

14.8.4 USART Control Register 1 (USARTx_CTLR1) (x=1/2)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		UE	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	RWU	SBK

Bit	Name	Access	Description	Reset value
[31:14]	Reserved	RO	Reserved	0
13	UE	RW	USART enable bit. When this bit is cleared, both the frequency divider and output of the USART will stop working after the current byte transfer is complete.	0
12	M	RW	Word length. 1: 9 data bits; 0: 8 data bits.	0
11	WAKE	RW	Wake-up position. This bit determines how to wake up USART. 1: Address mark 0: The bus is idle.	0
10	PCE	RW	Check bit enable. For the receiver, it is the parity of the data; for the sender, the parity bit is inserted. Once this bit is set, the parity bit enable will not take effect until the current byte transfer is complete.	0
9	PS	RW	Parity selection. 0 indicates even parity and 1 indicates odd parity. When this bit is set, the parity bit enable will not take effect until the current byte transfer is complete.	0
8	PEIE	RW	Parity check interrupt enable bit. This bit means that a parity error interrupt is allowed.	0
7	TXEIE	RW	Transmit buffer null interrupt enable. This bit indicates that a null interrupt in the transmit buffer is allowed.	0
6	TCIE	RW	Transmit completed interrupt enable. This position	0

			bit means that a transmission completion interrupt is allowed.	
5	RXNEIE	RW	The receive buffer is not null interrupt enable. This bit indicates that a non-null interrupt in the receive buffer is allowed.	0
4	IDLEIE	RW	Bus idle interrupt enable. This position bit means that a bus idle interrupt is allowed.	0
3	TE	RW	Transmit enable. Setting this bit will enable the transmitter.	0
2	RE	RW	Receive enable. Setting this bit enables the receiver, which begins to detect the start bit on the RX pin.	0
1	RWU	RW	Receive wake-up call. This bit determines whether to put the USART in silent mode. 1: The receiver is in silent mode; 0: The receiver is in normal operation mode. *Note 1: Before setting the RWU bit, the USART needs to receive a byte of data, otherwise it cannot be woken up idle by the bus in silent mode.* *Note 2: When configured to wake up with an address tag, the RWU bit cannot be modified by software when RXNE is set.*	0
0	SBK	RW	Transmit frame break character control bits. Set this bit to send a frame break character. When the stop bit of the frame is disconnected, it is reset by hardware. 1: Transmit; 0: Do not transmit.	0

14.8.5 USART Control Register 2 (USARTx_CTLR2) (x=1/2)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	LINEN	STOP[1:0]		Reserved					LBDIE	LBDL	Reserved	ADD[3:0]			

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RO	Reserved	0
14	LINEN	RW	LIN mode enable bit, setting enables LIN mode. In LIN mode, the LIN synchronization break symbol can be sent using the SBK bit and the LIN synchronization break can be detected.	0
[13:12]	STOP[1:0]	RW	Stop setting the domain. These two will set the	0

			stop bit. 00: 1 stop bit; 01: 0.5 stop bit; 10: 2 stop bits; 11: 1.5 stop bits.	
[11:7]	Reserved	RW	Reserved	0
6	LBDIE	RW	LIN Break detects interrupt enable, which enables interrupts caused by LBD.	0
5	LBDL	RW	LIN Break detects the length, which is used to select whether to detect an 11-bit or 10-bit breaker. 1: 11-bit breaker detection; 0: 10-bit breaker detection.	0
4	Reserved	RW	Reserved	0
[3:0]	ADD[3:0]	RW	Address domain, which is used to set the USART node address of this device. Used in silent mode in multiprocessor communication, using address tags to wake up a USART device.	0

14.8.6 USART Control Register 3 (USARTx_CTLR3) (x=1/2)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved					CTSIE	CTSE	RTSE	DMAT	DMAR	Reserved		HDSEL	IRLP	IREN	EIE

Bit	Name	Access	Description	Reset value
[31:11]	Reserved	RO	Reserved	0
10	CTSIE	RW	CTSIE interrupt enable bit, which will cause an interrupt when CTS is set.	0
9	CTSE	RW	CTS enable bit, which enables CTS flow control.	0
8	RTSE	RW	RTS enable bit, which enables RTS flow control.	0
7	DMAT	RW	DMA transmits enable bit. This location 1 uses DMA when transmitting.	0
6	DMAR	RW	The DMA receives enable bit. This location 1 uses DMA when receiving.	0
[5:4]	Reserved	RO	Reserved	0
3	HDSEL	RW	Half-duplex mode selection bit, set this bit to select half-duplex mode.	0
2	IRLP	RW	Infrared low-power selection bit, set this bit when selecting infrared, enable low-power mode.	0
1	IREN	RW	Infrared enable bit, set this bit to enable infrared mode.	0

0	EIE	RW	The error enables the interrupt bit. After setting this bit, if DMAR is set, if FE, ORE or NE is set, an interrupt will occur.	0

14.8.7 USART Prescaler Register (USARTx_GPR) (x=1/2)

Offset address: 0x18

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								PSC[7:0]							

Bit	Name	Access	Description	Reset value
[31:8]	Reserved	RO	Reserved	0
[7:0]	PSC[7:0]	RW	Prescaler range. In infrared low power mode, the source clock is divided by this value (Valid for all 8 bits), and a value of 0 indicates retention. In infrared normal mode, this bit can only be set to 1.	0

Chapter 15 Inter-integrated Circuit (I2C) interface

The module description in this chapter applies only to CH32V002, CH32V004, CH32V005, CH32V006, CH32V007, and CH32M007 products.

The internal integrated circuit bus (I2C) is widely used in the communication between microcontrollers, sensors and other off-chip modules. It supports multi-master and multi-slave mode, and can communicate at both 100kHz (Standard) and 400kHz (Fast) speeds using only two wires (SDA and SCL).

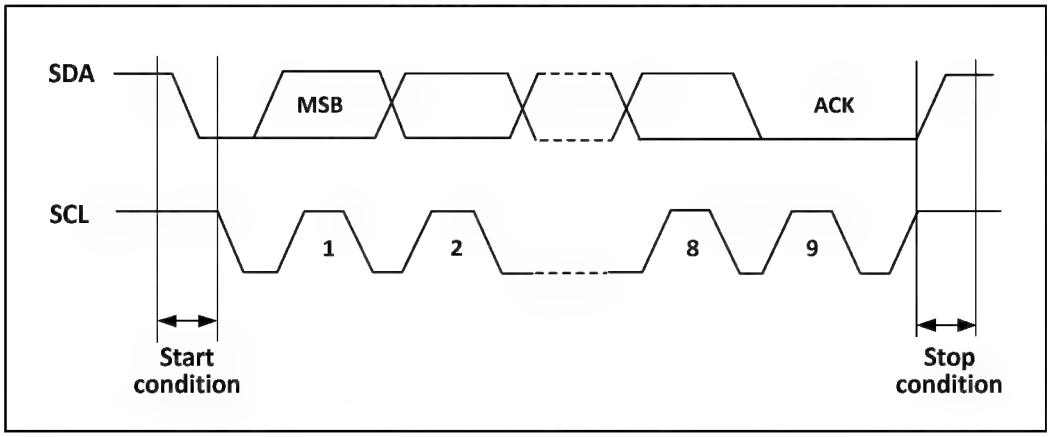
15.1 Main Features

- Support master and slave modes
- Support 7-bit or 10-bit addresses
- Slave devices support dual 7-bit addresses
- Support 2 speed modes: 100kHz and 400kHz
- Multiple status modes, multiple error flags
- Support extended clock function
- 2 interrupt vectors
- DMA support
- Support PEC

15.2 Overview

I2C is a half-duplex bus, which can only run in one of the following four modes: master device transmitting mode, master device receiving mode, slave device transmitting mode and slave device receiving mode. The I2C module works in slave mode by default. After generating the starting condition, it will automatically switch to the master mode. When the arbitration is lost or a stop signal is generated, it will switch to the slave mode. The I2C module supports multi-host functions. When working in master mode, the I2C module will actively send out data and addresses. Data and addresses are transmitted in 8-bit units, with the high bit in front and the low bit in the back. After the start event is a byte (7-bit address mode) or two-byte (10-bit address mode) address. Every time the host sends 8-bit data or address, the slave needs to reply a reply ACK, that is, pull down the SDA bus, as shown in figure 15-1.

Figure 15-1 I2C Timing Diagram

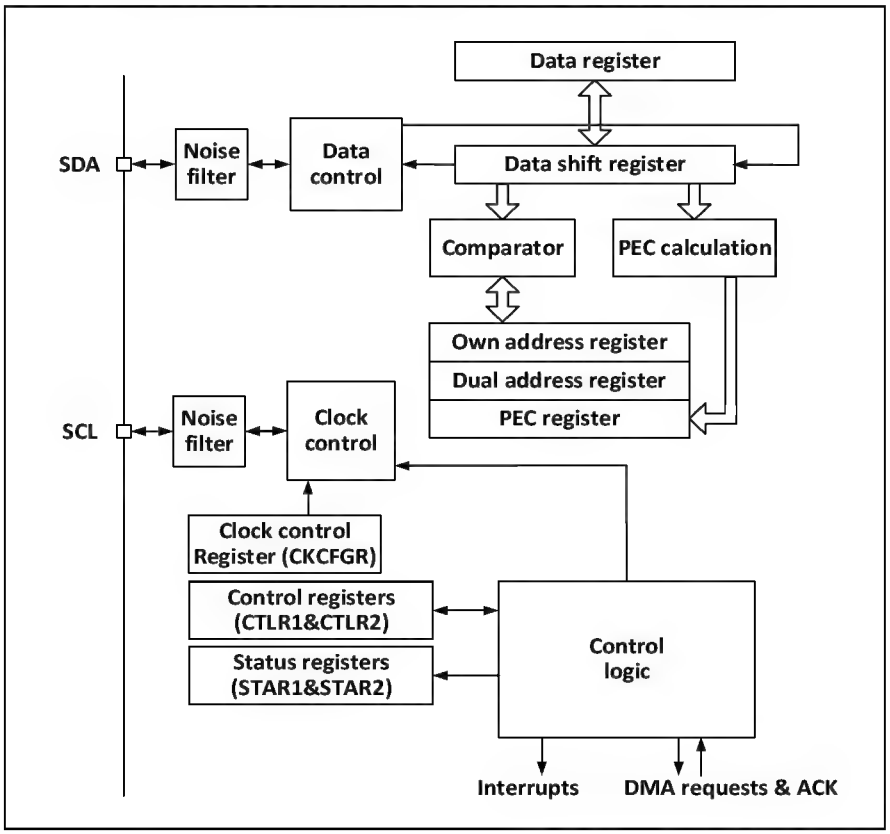


In order to use it normally, the correct clock must be input to I2C. In standard mode, the lowest input clock is 2MHz,

and in fast mode, the lowest input clock is 4MHz.

Figure 15-2 is the functional block diagram of the I2C module.

Figure 15-2 Functional block diagram of the I2C module



15.3 Master Mode

In master mode, the I2C module dominates the data transmission and outputs the clock signal. The data transmission begins with the start event and ends with the end event. The steps to communicate using main mode are:

Set the correct clock in the control register 2 (R16_I2C1_CTLR2) and the clock control register (R16_I2C1_CKCFGR);

Set the appropriate rising edge in the rising edge register (R16_I2C1_RTR);

Set the PE bit in the control register (R16_I2C1_CTLR1) to start the peripheral;

Set the START bit in the control register (R16_I2C1_CTLR1) to generate a start event.

After setting the START bit, the I2C module will automatically switch to the main mode, the MSL bit will be set, and the starting event will be generated. After the initial event is generated, the SB bit will be set, and if the ITEVTEN bit (in R16_I2C1_CTLR2) is set, it will be interrupted. At this point, the status register 1 (R16_I2C1_STAR1) should be read. After writing from the address to the data register, the SB bit will be cleared automatically.

If the 10-bit address mode is used, then write the data register send header sequence (the header order is listed as 11110xx0b, where the xx bit is the highest two bits of the 10-bit address).

After sending the header sequence, the ADD10 bit of the status register will be set, and if the ITEVTEN bit has been set, an interrupt will occur. After reading the R16_I2C1_STAR1 register, write the second address byte to the data register, and clear the ADD10 bit.

Then write the data register to send the second address byte. After sending the second address byte, the ADDR bit

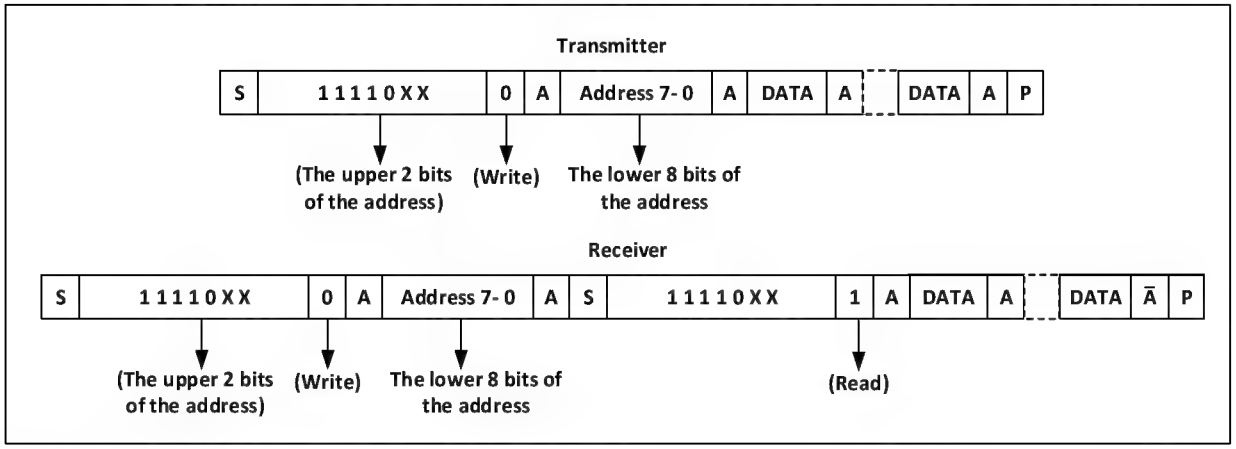
of the status register will be set. If the ITEVTEN bit has been set, it will cause an interrupt. In this case, you should read the R16_I2C1_STAR1 register and read the R16_I2C1_STAR2 register again to clear the ADDR bit.

If the 7-bit address mode is used, the write data register sends address bytes. After the address bytes are sent, the ADDR bit of the status register will be set, and if the ITEVTEN bit has been set, an interrupt will occur. At this time, you should read the R16_I2C1_STAR1 register and read the R16_I2C1_STAR2 register again to clear the ADDR bit.

In 7-bit address mode, the first byte sent is the address byte, the first 7 bits represent the address of the target slave device, the 8th bit determines the direction of the subsequent message, 0 indicates that the master device writes data to the slave device, and 1 represents that the master device reads information from the slave device.

In the 10-bit address mode, as shown in Figure 15-3, during the send address phase, the first byte is 11110xx0, xx is the highest 2 bits of the 10-bit address, and the second byte is the lower 8 bits of the 10-bit address. If you subsequently enter the master device transmit mode, you will continue to send data; if you subsequently prepare to enter the master device receive mode, you will need to re-send a start condition to follow the sending of a byte as 11110xx1, and then enter the master device receive mode.

Figure 15-3 Schematic diagram of master transmitting and receiving data at 10-bit address

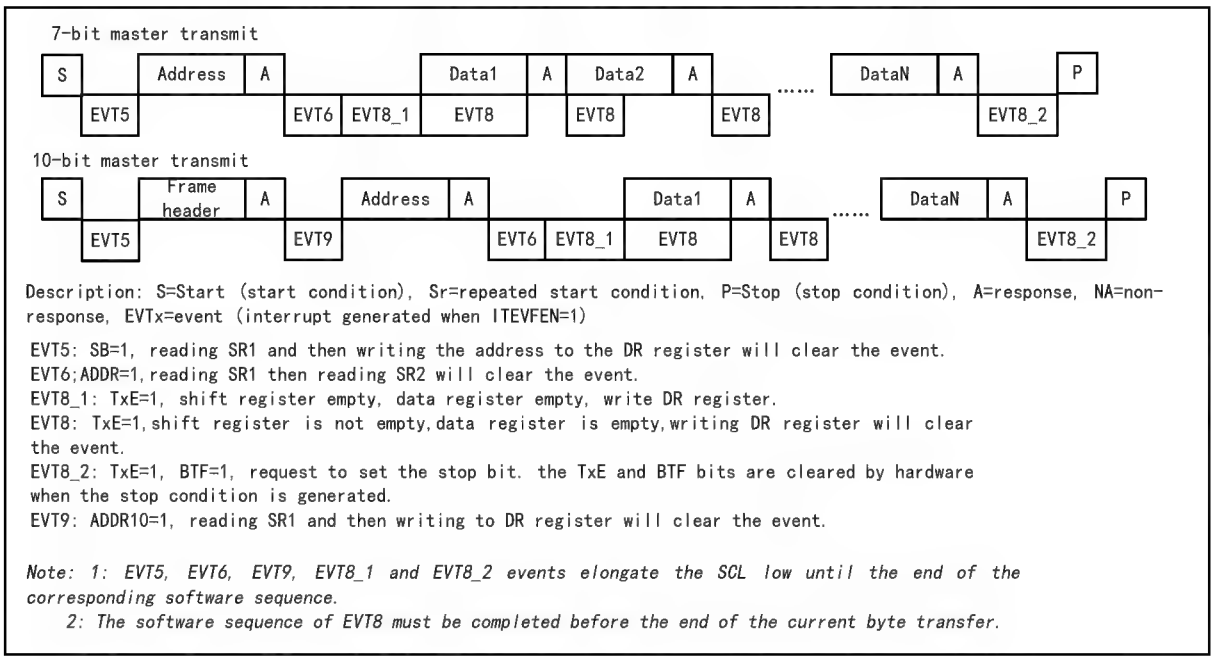


Master transmit mode:

The shift register inside the main device transmits data from the data register to the SDA line. When the master device receives the ACK, the TxE of the status register 1 (R16_I2C1_STAR1) is set, and an interrupt occurs if the ITEVTEN and ITBUFEN are set. Writing data to the data register clears the TxE bit.

If the TxE bit is set and no new data is written to the data register before the last data is sent, then the BTF bit will be set, the SCL will remain low until it is cleared, and after reading the R16_I2C1_STAR1, writing data to the data register will clear the BTF bit.

Figure 15-4 Master transmitter transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

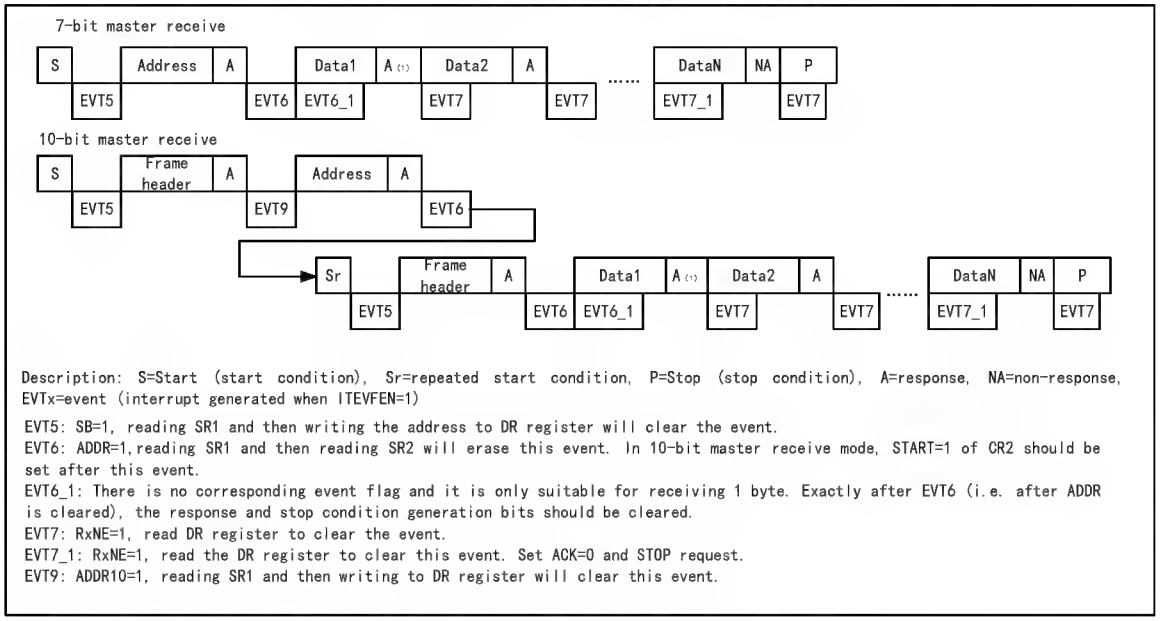
EVT5: SB=1, reading SR1 and then writing the address to the DR register will clear the event.
EVT6;ADDR=1,reading SR1 then reading SR2 will clear the event.
EVT8_1: TxE=1, shift register empty, data register empty, write DR register.
EVT8: TxE=1,shift register is not empty,data register is empty,writing DR register will clear the event.
EVT8_2: TxE=1, BTF=1, request to set the stop bit. the TxE and BTF bits are cleared by hardware when the stop condition is generated.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear the event.

Note: 1: EVT5, EVT6, EVT9, EVT8_1 and EVT8_2 events elongate the SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT8 must be completed before the end of the current byte transfer.

Master receive mode:

The I2C module receives data from the SDA line and writes it into the data register through the shift register. After each byte, if the ACK bit is set, the I2C module will issue a low level of reply, while the RxNe bit will be set, and if ITEVTEN and ITBUFEN are set, there will be an interruption. If the RxNE is set and the original data is not read before the new data is received, the BTF bit will be set, the SCL will remain low before clearing the BTF, and reading the R16_I2C1_STAR1 and then reading the data register will clear the BTF bit.

Figure 15-5 Receiver transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT5: SB=1, reading SR1 and then writing the address to DR register will clear the event.
EVT6: ADDR=1,reading SR1 and then reading SR2 will erase this event. In 10-bit master receive mode, START=1 of CR2 should be set after this event.
EVT6_1: There is no corresponding event flag and it is only suitable for receiving 1 byte. Exactly after EVT6 (i.e. after ADDR is cleared), the response and stop condition generation bits should be cleared.
EVT7: RxNE=1, read DR register to clear the event.
EVT7_1: RxNE=1, read the DR register to clear this event. Set ACK=0 and STOP request.
EVT9: ADDR10=1, reading SR1 and then writing to DR register will clear this event.

When the master device ends sending data, it will actively send an end event, that is, set the STOP bit, and I2C will switch to slave mode. In the receive mode, the master device needs to NAK at the answer position of the last data bit. After receiving the NACK, the slave device releases control over the SCL and SDA lines; the master device can send a stop / restart condition. Note that after the stop condition is generated, the I2C module will automatically

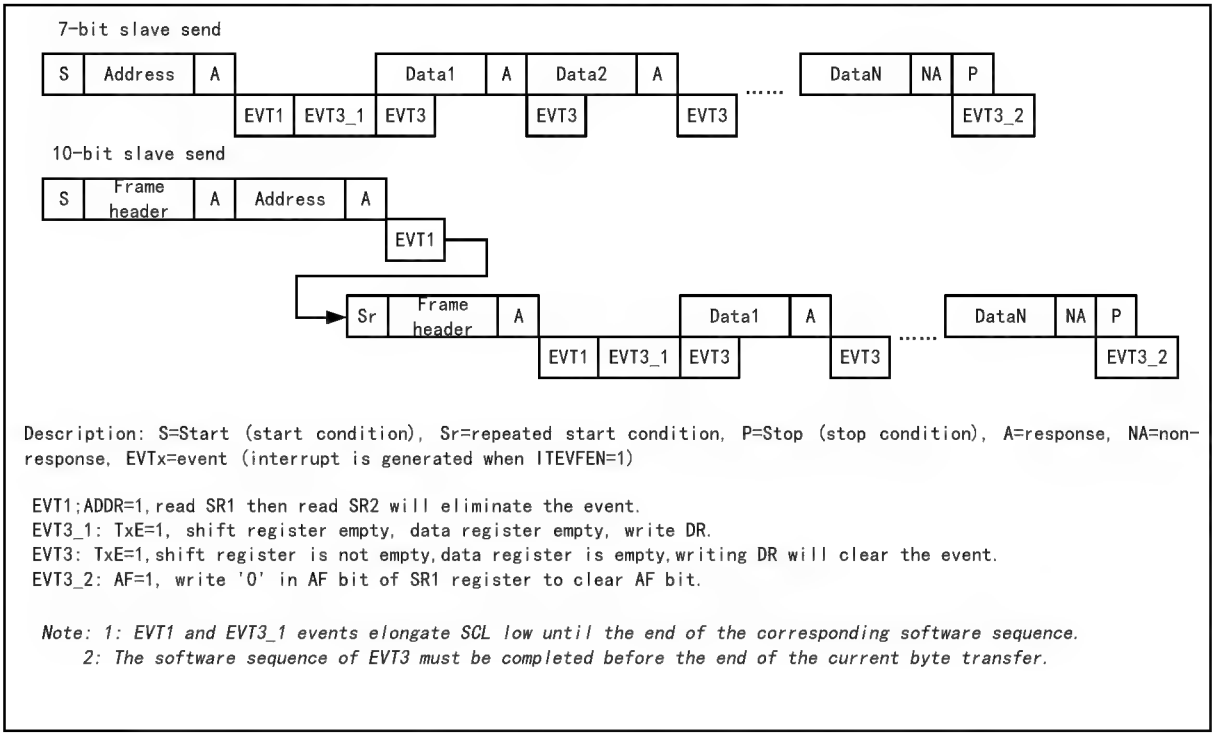
switch to slave mode.

15.4 Slave Mode

In the salve mode, the I2C module can recognize its own address and broadcast call address. The software can control the identification of broadcast call addresses on or off. Once the start event is detected, the I2C module compares the SDA data through the shift register with its own address (the number of bits depends on ENDUAL and ADDMODE) or the broadcast address (when ENGC is set). If it does not match, it will ignore it until a new start event is generated; if it matches the header sequence, it will generate an ACK signal and wait for the address of the second byte. If the address of the second byte also matches or the entire segment of the address matches in the case of a 7-bit address, then:
First generate an ACK response;
The ADDR bit is set, and if the ITEVTEN bit is set, there will be a corresponding interrupt
If you are using dual-address mode (the ENDUAL bit is set), you also need to read the DUALF bit to determine which address the host is calling.
The slave mode defaults to receive mode, after the last bit of the received header sequence is 1, or the last bit of the 7-bit address is 1 (Depending on whether the header sequence is received for the first time or a normal 7-bit address), when a duplicate starting condition is received, the I2C module will enter sender mode, and the TRA bit will indicate whether it is currently in receiver or sender mode.

Slave transmit mode:
After clearing the ADDR bit, the I2C module transmits bytes from the data register to the SDA line through the shift register. The slave device maintains the SCL at a low level until the ADDR bit is cleared and the data to be transmitted has been written to the data register. (see EVT1 and EVT3 in the following figure). Upon receipt of a reply ACK, the TXE bit is set, and an interrupt occurs if ITEVTEN and ITBUFEN are set. If TxE is set but no new data is written to the data register before the end of the next data transmission, the BTF bit will be set. The SCL will remain low until the BTF is cleared, and after reading the status register 1 (R16_I2C1_STAR1), writing data to the data register will clear the BTF bit.

Figure 15-6 Slave transmitter transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt is generated when ITEVFEN=1)

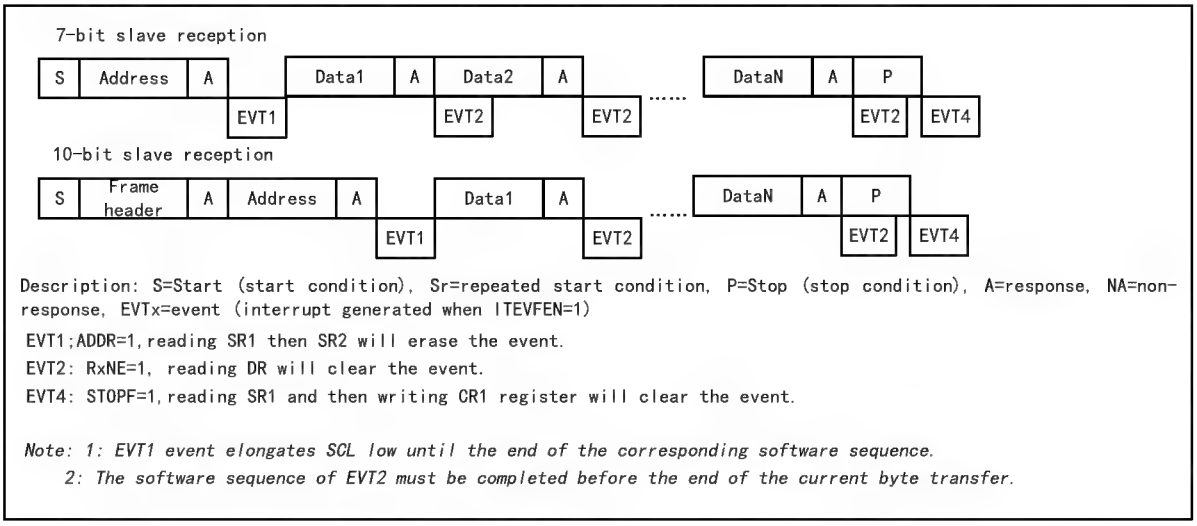
EVT1;ADDR=1,read SR1 then read SR2 will eliminate the event.
EVT3_1: TxE=1, shift register empty, data register empty, write DR.
EVT3: TxE=1,shift register is not empty,data register is empty,writing DR will clear the event.
EVT3_2: AF=1, write '0' in AF bit of SR1 register to clear AF bit.

Note: 1: EVT1 and EVT3_1 events elongate SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT3 must be completed before the end of the current byte transfer.

Slave receive mode:

After the ADDR is cleared, the I2C module stores the data on the SDA into the data register through the shift register. After each byte is received, the I2C module will set an ACK bit and a RxNE bit. If ITEVTEN and ITBUFEN are set, an interrupt is also generated. If the RxNE is set and the old data is not read out before the new data is received, then the BTF will be set. The SCL stays low until the BTF bit is cleared. Reading the status register 1 (R16_I2C1_STAR1) and reading the data in the data register clears the BTF bit.

Figure 15-7 Receiver transmission sequence diagram



Description: S=Start (start condition), Sr=repeated start condition, P=Stop (stop condition), A=response, NA=non-response, EVTx=event (interrupt generated when ITEVFEN=1)

EVT1;ADDR=1,reading SR1 then SR2 will erase the event.
EVT2: RxNE=1, reading DR will clear the event.
EVT4: STOPF=1,reading SR1 and then writing CR1 register will clear the event.

Note: 1: EVT1 event elongates SCL low until the end of the corresponding software sequence.
2: The software sequence of EVT2 must be completed before the end of the current byte transfer.

After the master device transmits the last data byte, it will produce a stop condition. When the I2C module detects the stop event, it will set the STOPF bit, and if the ITEVFEN bit is set, it will also produce an interrupt. The user needs to read the status register (R16_I2C1_STAR1) and then write the control register (Such as the reset control word SWRST) to clear. (See EVT4 in the figure above).

15.5 Error Conditions

15.5.1 Bus Error (BERR)

When the I2C module detects an external start or stop event during address or data transmission, a bus error occurs. When a bus error is generated, the BERR bit is set, and an interrupt occurs if ITERREN is set. In slave mode, the data is discarded and the hardware releases the bus. If it is a start signal, the hardware will consider it as a restart signal and start waiting for the address or stop signal; if it is a stop signal, it will operate according to the normal stop condition in advance. In master mode, the hardware does not release the bus and does not affect the current transmission, and it is up to the user code to decide whether to abort the transmission.

15.5.2 Acknowledge Failure (AF)

When the I2C module detects no response after a byte, a reply error occurs. When a reply error is generated: AF will be set, and an interrupt will be generated if ITERREN is set; if an AF error is encountered, if the I2C module works in slave mode, the hardware must release the bus, and if it is in master mode, the software must generate a stop event.

15.5.3 Arbitration Lost (ARLO)

When the I2C module detects the arbitration loss, an arbitration loss error occurs. When an arbitration loss error occurs: the ARLO bit is set and an interrupt is generated if ITERREN is set; the I2C module switches to slave mode and no longer responds to slave-initiated transfers against it unless a host initiates a new start event; the hardware releases the bus.

15.5.4 Overrun/underrun Error (OVR)

- **Overrun error**

In slave mode, if clock extension is prohibited, the I2C module is receiving data. If one byte of data has been received, but the last received data has not been read out, an overload error will occur. When an overload error occurs, the last received byte is discarded and the sender should resend the last sent byte.

- **Underrun error**

In slave mode, if clock extension is prohibited, the I2C module is sending data, and if new data is not written to the data register before the next byte clock arrives, an underload error will occur. When an underload error occurs, the data in the previous data register will be sent twice. If an underload error occurs, the receiver should discard the repeatedly received data. In order not to generate underload errors, the I2C module should write data to the data register before the first rising edge of the next byte.

15.6 Clock Extension

If clock extension is prohibited, there is a possibility of overload / underload errors. But if the clock is extended when enabled:

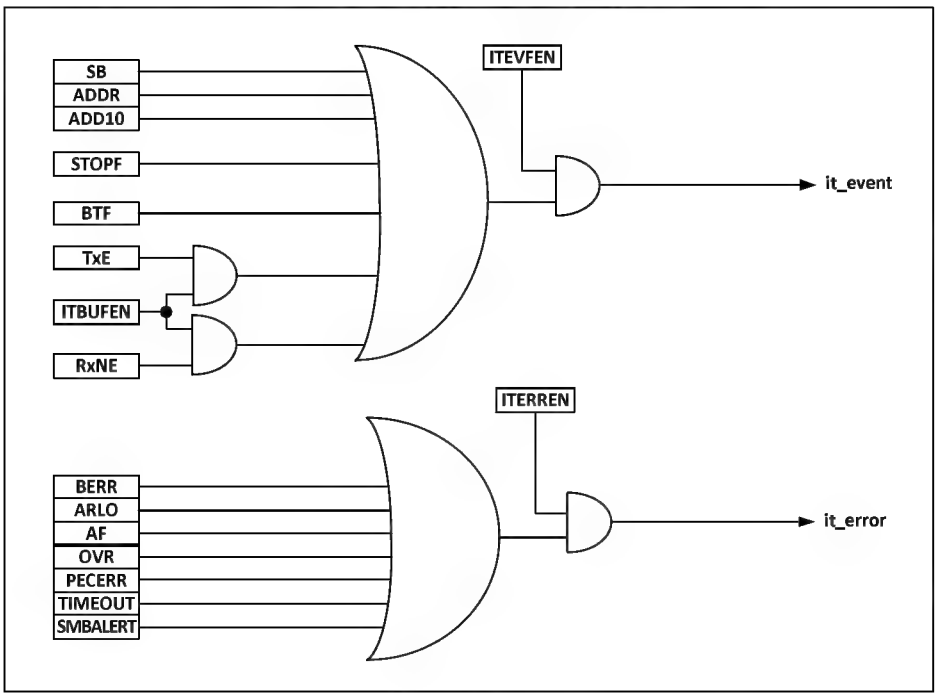
- In transmit mode, if TxE is set and BTF is set, the SCL will remain low, waiting for the user to read the status register and write the data to be sent to the data register;
- In receive mode, if RxNE is set and BTF is set, the SCL will remain low after receiving the data until the user reads the status register and reads the data register.

Thus it can be seen that the extension of the enable clock can avoid overload / underload errors.

15.7 Interrupts

Each I2C module has two kinds of interrupt vectors, which are event interrupt and error interrupt. The two interrupts support the interrupt source in figure 15-8.

Figure 15-8 I2C Interrupt Request



15.8 DMA

You can use DMA to send and receive bulk data. The ITBUFEN bit of the control register cannot be set when using DMA.

- Use DMA to transmit

The DMA mode can be activated by setting the DMAEN position bit of the CTLR2 register. As long as the TxE bit is set, the data will be loaded by the DMA from the set memory into the I2C data register. The following settings are required to assign channels to I2C.

1) The I2C1_DATAR register address is set to the DMA_PADDRx register, and the memory address is set in the DMA_MADDRx register, so that after each TxE event, data is sent from the memory to the I2C1_DATAR register.
2) Set the required number of transfer bytes in the DMA_CNTRx register. This value is decremented after each TxE event.
3) Configure the channel priority using the PL[0:1] bits in the DMA_CFGRx register.
4) Set the DIR bit in the DMA_CFGRx register, and according to the application requirements, you can configure to issue an interrupt request when the whole transmission is half or all complete.
5) Activate the channel by setting the EN bit on the DMA_CFGRx register.

When the number of data transmission bytes set in the DMA controller has been completed, the DMA controller sends an EOT/EOT_1 signal at the end of the transmission to the I2C interface. If the interrupt allows, an DMA interrupt will be generated.

- Use DMA to receive

After setting the DMAEN of the CTLR2 register, the DMA receiving mode can be carried out. When receiving using DMA, DMA transfers the data in the data register to a preset memory area. The following steps are required to assign channels to I2C.

1) The I2C1_DATAR register address is set to the DMA_PADDRx register, and the memory address is set in the DMA_MADDRx register, so that after each RxNE event, the data is written from the I2C1_DATAR register to the memory.
2) Set the required number of transfer bytes in the DMA_CNTRx register. This value is decremented after each RxNE event.
3) Configure the channel priority with PL [0:1] in the DMA_CFGRx register.
4) Clear the DIR bit in the DMA_CFGRx register, and according to the application requirements, it can be set to issue an interrupt request when half or all of the data transmission is completed.
5) Set the EN bit in the DMA_CFGRx register to activate the channel.

When the number of data transmission bytes set in the DMA controller has been completed, the DMA controller sends an EOT/EOT_1 signal at the end of the transmission to the I2C interface. If the interrupt allows, an DMA interrupt will be generated.

15.9 Packet Error Checking

Packet error check (PEC) is a step to add a CRC8 check to provide transmission reliability, using the following polynomials to calculate each bit of serial data:

$$C=X^8+X^2+X+1$$

PEC computing is activated by the ENPEC bit of the control register and calculates all bytes of information, including address and read-write bits. When sending, enabling PEC will add a CRC8 calculation result after the last byte of data; while in the receive mode, the last byte is considered to be the CRC8 check result, and if it does not match the internal calculation result, it will reply a NAK. If it is the main receiver, it will reply a NAK regardless of whether the check result is correct or not.

15.10 Register Description

Table 15-1 I2C-related registers list

Name	Offset address	Description	Reset value
R16_I2C1_CTLR1	0x40005400	I2C Control Register 1	0x0000
R16_I2C1_CTLR2	0x40005404	I2C Control Register 2	0x0000
R16_I2C1_OADDR1	0x40005408	I2C Address Register 1	0x0000
R16_I2C1_OADDR2	0x4000540C	I2C Address Register 2	0x0000
R16_I2C1_DATAR	0x40005410	I2C Data Register	0x0000
R16_I2C1_STAR1	0x40005414	I2C Status Register 1	0x0000
R16_I2C1_STAR2	0x40005418	I2C Status Register 2	0x0000
R16_I2C1_CKCFGR	0x4000541C	I2C Clock Register	0x0000

15.10.1 I2C Control Register 1(I2C1_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWRST	Reserved		PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	Reserved				PE

Bit	Name	Access	Description	Reset value
15	SWRST	RW	Software reset, user code setting this bit will cause I2C peripherals to reset. Before resetting, it is determined that the pin of the I2C bus is released and the bus is idle. *Note: This bit can reset the I2C module when the stop condition is not detected on the bus but the bus bit is 1.*	0
[14:13]	Reserved	RO	Reserved	0
12	PEC	RW	Packet error detection enable bit, set this bit to enable packet error detection. The user code can zero or zero this bit; when the PEC is transmitted, it generates a start or end signal, or when the PE bit is cleared, the hardware clears the bit. 1: With PEC; 0: Without PEC. *Note: PEC is invalid when arbitration is lost.*	0
11	POS	RW	The ACK and PEC position setting bit, which can be set or cleared by user code, can be cleared by hardware after PE is cleared. 1: The ACK bit controls the ACK or NAK of the next byte received in the shift register. The next byte received in the PEC shift register is PEC; 0: The ACK bit controls the ACK or NAK of the bytes being accepted in the current shift register. The PEC bit indicates that the byte of the pre-bit shift register is PEC. *Note: The use of the POS bit in 2-byte data reception is as follows: it must be configured before receiving. For the second byte of NACK, the ACK bit must be cleared immediately after clearing the ADDR bit; in order to detect the PEC of the second byte, the PEC bit must be set after the POS bit is configured after the ADDR event occurs.*	0
10	ACK	RW	Reply enable bit, which can be set or zeroed by the user code, and when the PE bit is set, it can be	0

			cleared by hardware. 1: Returns a reply after receiving a byte; 0: No answer is set.	
9	STOP	RW	Stop event generation bit, the bit can be set or zeroed by the user code, or cleared by the hardware when a stop event is detected, or set by the hardware when a timeout error is detected. In master mode: 1: A stop event occurs after the current byte transfer or the current start condition is issued; 0: No stop event occurs. In slave mode: 1: Release SCL and SDA lines after the current byte transfer; 0: No stop event occurs.	0
8	START	RW	Start event generation bit, The bit can be set or zeroed by the user code, and zeroed by the hardware when the initial condition is issued or the PE is cleared at 00:00. In master mode: 1: Repeat the start event 0: No start event occurs. In slave mode: 1: When the bus is idle, a start event is generated 0: No start event occurs.	0
7	NOSTRETCH	RW	Disable clock extension bit, this bit is used to prevent the clock from being extended from the mode until it is cleared by the software when the ADDR or BTF flag is set. 1: Disable clock extension; 0: Enable clock extension.	0
6	ENGC	RW	Broadcast call enable bit, set this bit to enable broadcast call, answer broadcast address 00h.	0
5	ENPEC	RW	PEC enable bit, set this bit to turn on PEC calculation.	0
[4:1]	Reserved	RO	Reserved	0
0	PE	RW	I2C peripheral enable bit. 1: Enable I2C module; 0: Disable I2C module.	0

15.10.2 I2C Control Register 2(I2C1_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			LAST	DMA	ITBU	ITEV	ITER	Reserved		FREQ[5:0]					

		EN	FEN	TEN	REN		

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	LAST	RW	DMA the last transfer of the set bit. 1: The next EOT of DMA is the last transmission; 0: The next EOT of DMA is not the last transmission. *Note: This bit is used in the main receive mode and can generate a NAK the last time the data is received.*	0
11	DMAEN	RW	DMA request enable bit, which allows DMA requests when TxE or RxEN is set.	0
10	ITBUFEN	RW	Buffer interrupt enable bit. 1: An event interrupt occurs when TxE or RxEN is set; 0: When TxE or RxEN is set, no interruption occurs.	0
9	ITEVTEN	RW	Event interrupt enable bit, set this bit to enable event interruption. This interrupt occurs under the following conditions: SB=1 (Master mode); ADDR=1 (Master/slave mode); ADDR10=1 (Master mode); STOPF=1 (Slave mode); BTF=1, but there is no TxE or RxEN event; If ITBUFEN=1; TxE event is 1; If ITBUFEN=1, RxNE event is 1.	0
8	ITERREN	RW	Error interrupt enable bit, Setting indicates the error interrupt is allowed. The interrupt occurs under the following conditions: BERR=1; ARLO=1; AF=1; OVR=1; PECERR=1; TIMEOUT=1; SMBAlert=1.	0
[7:6]	Reserved	RO	Reserved	0
[5:0]	FREQ[5:0]	RW	I2C module clock frequency domain, must input the correct clock frequency to produce the correct timing, the allowable range is between 8-48MHz. Must be set between 001000b and 110000b in MHz.	0

15.10.3 I2C Address Register 1(I2C1_OADDR1)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDMODE	Reserved							ADD[9:8]		ADD[7:1]					ADD0

Bit	Name	Access	Description	Reset value
15	ADDMODE	RW	Address mode. 1: 10-bit slave address (Does not respond to 7-bit address); 0: 7-bit slave address (Does not respond to 10-bit address);	0
[14:10]	Reserved	RO	Reserved	0
[9:8]	ADD[9:8]	RW	Interface address, which is 9-8 bits when using 10-bit addresses and ignored when 7-bit addresses are used.	0
[7:1]	ADD[7:1]	RW	Interface address, bits 7-1.	0
0	ADD0	RW	Interface address, which is bit 0 when a 10-bit address is used and ignored when a 7-bit address is used.	0

15.10.4 I2C Address Register 2(I2C1_OADDR2)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								ADD2[7:1]							ENDUAL

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
[7:1]	ADD2[7:1]	RW	Interface address, 7-1 bits of the address in dual-address mode.	0
0	ENDUAL	RW	Dual address mode enable bit, set this bit so that ADD2 can also be recognized.	0

15.10.5 I2C Data Register (I2C_DATAR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								DR[7:0]							

Bit	Name	Access	Description	Reset value
15:8	Reserved	RO	Reserved	0

7:0	DR[7:0]	RW	Data register, which is used to store received data or data sent to the bus.	0

15.10.6 I2C Status Register 1(I2C_STAR1)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB

Bit	Name	Access	Description	Reset value
[15:13]	Reserved	RO	Reserved	0
12	PECERR	RW0	A PEC error flag bit occurs during reception, which can be reset by user write 0 or by hardware when the PE becomes low. 1: There is a PEC error. After receiving the PEC, return NAK; 0: No PEC errors.	0
11	OVR	RW0	Overload and underload mark bit. 1: Overload and underload events occur: when a new byte is received in the receive mode in NOSTRETCH=1, the newly received byte will be lost if the contents of the data register are not read out; in the send mode, no new data is written to the data register, and the same byte will be sent twice; 0: There are no overload or underload events.	0
10	AF	RW0	Reply failure flag bit, which can be reset by user write 0 or by hardware when the PE becomes low. 1: Answer error 0: The answer is normal.	0
9	ARLO	RW0	Arbitration loses the flag bit, which can be reset by user write 0 or by hardware when the PE becomes low. 1: Loss of arbitration was detected and the module lost control of the bus 0: Arbitration is normal.	0
8	BERR	RW0	Bus error flag bit, This bit can be reset by user write 0 or by hardware when the PE becomes low. 1: Error in start or stop condition 0: Normal.	0
7	TxE	RO	The data register is an empty log bit, and writing data to the data register can be cleared, or after a start or stop bit is generated, or when the PE is 0, it is automatically cleared by the hardware. 1: When transmitting data, the transmit data	0

			register is empty 0: The data register is not empty.	
6	RxNE	RO	The data register is not an empty log bit, which will be cleared by reading and writing to the data register, or by the hardware when PE is 0. 1: When receiving data, the data register is not empty; 0: Normal.	0
5	Reserved	RO	Reserved	0
4	STOPF	RO	Stop the event flag bit, after the user reads the status register 1, the write to the control register 1 will clear this bit, or when the PE is 0, the hardware will clear this bit. 1: After the reply, a stop event is detected on the bus from the device; 0: No stop event was detected.	0
3	ADD10	RO	The 10-bit address header sequence sends flag bits. After the user reads status register 1, the write operation to control register 1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: In 10-bit address mode, the master device has sent the first address byte; 0: None.	0
2	BTF	RO	Byte transmission end flag bit. After the user reads status register 1, the read and write to the data register clears this bit; in transmission, after initiating a start or stop event, or when PE is 0, the hardware clears this bit. 1: End of byte transmission. When NOSTRETCH=0: when sending, when a new data is sent and the data register has not been written to the new data; when receiving, when a new byte is received but the data register has not been read; 0: None.	0
1	ADDR	RW0	The address is sent / the address matches the flag bit. After the user reads the status register 1, the read operation on the status register 2 will clear this bit, or when the PE is 0, the hardware will clear this bit. Master mode: 1: End of address sending: in 10-bit address mode, it is set when the second byte ACK of the address is received; in 7-bit address mode, it is set when the ACK of the address is received.	0

			0: The address transmission did not end. Slave mode: 1: The received address matches; 0: The address does not match or the address was not received.	
0	SB	RO	Start bit transmit flag bit, and the operation of writing the data register after reading status register 1 will clear this bit, or when PE is 0, the hardware will clear this bit. 1: Start bit sent; 0: The start bit is not sent.	0

15.10.7 I2C Status Register 2 (I2C_STAR2)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEC[7:0]								DUALF	Reserved		GENCALL	Reserved	TRA	BUSY	MSL

Bit	Name	Access	Description	Reset value
[15:8]	PEC[7:0]	RO	Packet error check domain, which holds the value of PEC when PEC is enabled (ENPEC is set).	0
7	DUALF	RO	Match detection flag bit, which is cleared by the hardware when a stop bit or start bit is generated, or when it is in PE=0. 1: The received address matches the content in OAR2; 0: The received address matches the content in the OAR1.	0
[6:5]	Reserved	RO	Reserved	0
4	GENCALL	RO	Broadcast call address flag bit, which is cleared by the hardware when a stop bit or start bit is generated, or when it is in PE=0. 1: When ENGC=1, the address of the broadcast call is received; 0: The broadcast call address was not received.	0
3	Reserved	RO	Reserved	0
2	TRA	RO	Transmit / receive flag bits that are cleared by the hardware when a stop event (STOPF=1), duplicate start condition, bus arbitration loss (ARLO=1), or PE=0 is detected. 1: Data transmitted; 0: Data received. This bit is determined based on the Rbank W bit	0

			of the address byte.	
1	BUSY	RO	Bus busy flag bit, which is cleared when a stop bit is detected. This information is still updated when the interface is disabled (PE=0). 1: Bus busy: low level exists in SDA or SCL; 0: The bus is idle and has no communication.	0
0	MSL	RO	Master-slave mode indicator bit. When the interface is in master mode (SB=1), the hardware sets the bit; when the bus detects a stop bit, when the arbitration is missing, or when PE=0, the hardware clears the bit.	0

15.10.8 I2C Clock Register (I2C1_CKCFGR)

Offset address: 0x1C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F/S	DUTY	Reserved		CCR[11:0]											

Bit	Name	Access	Description	Reset value
15	F/S	RW	Master mode selection bit. 1: Fast mode; 0: Standard mode.	0
14	DUTY	RW	Duty cycle in fast mode: 1: $T_{Low}/T_{High} = 16/9$; 0: $T_{Low}/T_{High} = 2$.	0
[13:12]	Reserved	RO	Reserved	0
[11:0]	CCR[11:0]	RW	Clock frequency division coefficient domain, determine the frequency waveform of the SCL clock.	0

Chapter 16 Serial Peripheral Interface (SPI)

The module description in this chapter applies only to CH32V002, CH32V004, CH32V005, CH32V006, CH32V007, and CH32M007 products.

SPI supports data exchange in three-wire synchronous serial mode, plus chip line selection supports hardware switching between master and slave modes, and supports communication with a single data line.

16.1 Main Features

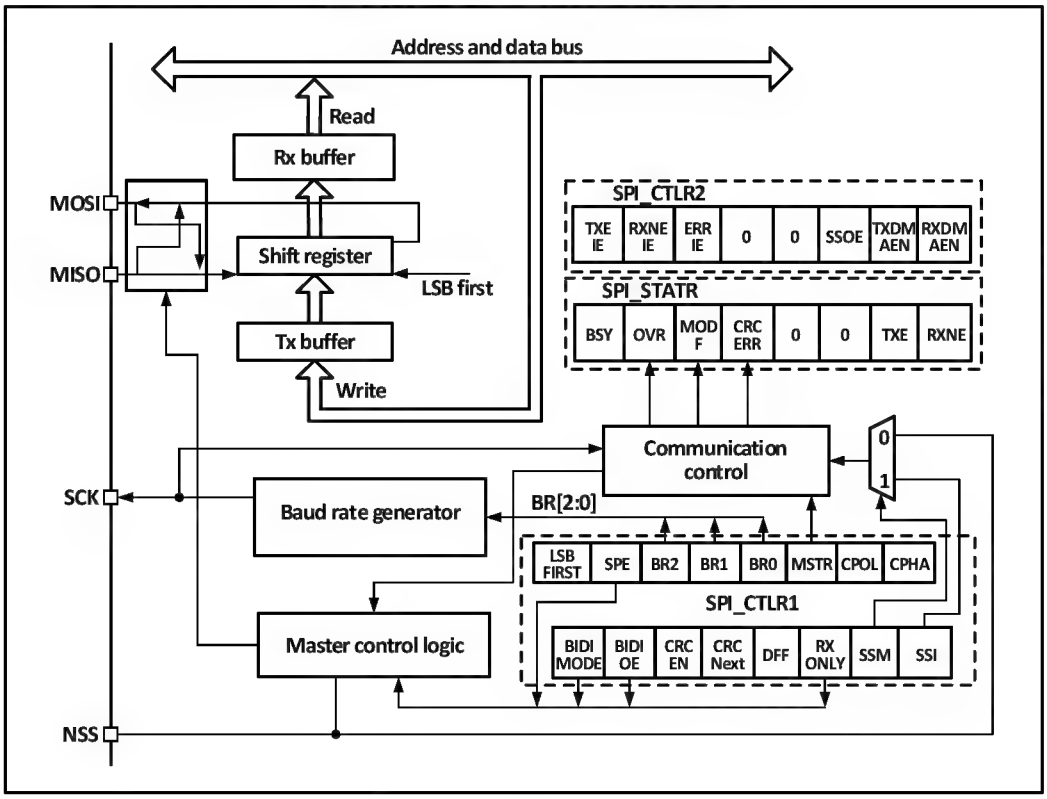
16.1.1 SPI Features

- Support full-duplex synchronous serial mode
- Support single-wire half-duplex mode
- Support Master mode and Slave mode, multi-slave mode
- Support 8-bit or 16-bit data structures
- Maximum clock frequency supports up to half of F_{HCLK}
- Data order supports MSB or LSB first
- Support hardware or software control of NSS pins
- Hardware CRC checksum support for sending and receiving
- Transceiver buffers support DMA transfers
- Support modification of clock phase and polarity

16.2 Function Description

16.2.1 Overview

Figure 16-1 SPI structure block diagram



As can be seen from Figure 16-1, the main SPI-related pins are MISO, M0SI, SCK and NSS. The MISO pin is a data input pin when the SPI module is working in master mode; it is a data output pin when it is working in slave mode. The MOSI pin is a data output pin when the SPI module is working in master mode; it is a data input pin when it is working in slave mode. The SCK is a clock pin. The clock signal is always output from the host, and the slave receives the clock signal and synchronizes the data sending and receiving. The NSS pin is a chip select pin and has the following usage:

1) NSS is controlled by software: When SSM is set, the internal NSS signal is determined by SSI whether the output is high or low. This case is generally used in SPI main mode.
2) NSS is controlled by hardware: When NSS output is enabled, that is, when SSOE is set, when the SPI host sends output, it will actively pull down the NSS pin. If the NSS pin is not successfully pulled, it means that other master devices on the main line are communicating, and a hardware error will occur; if SSOE is not set, it can be used in multi-host mode. If it is pulled down, it will be forced to enter the slave mode, and the MSTR bit will be automatically cleared.

The operating mode of SPI can be configured through CPHA and CPOL. The CPHA setting indicates that the module samples the data at the second edge of the clock, and the data is latched, while the CPHA setting indicates that the SPI module samples at the first edge of the clock, and the data is latched. CPOL indicates whether the countless data clock is high or low.

The host and device need to be set to the same SPI mode, and the SPE bit needs to be cleared before configuring SPI mode. The DEF bit can determine whether the single data length of the SP is 8 or 16 bits. LSBFIRST can control whether a single data word is high or low.

Table 16-1 SPI schema differentiation

	Mode 0	Mode 1	Mode 2	Mode 3
CPOL	0	1	1	1
CPHA	0	0	0	1

16.2.2 Master Mode

When the SPI module works in the main mode, the serial clock is generated by the SCK. Configure to main mode to perform the following steps:

Configure the BR [2:0] field of the control register to determine the clock;

Configure the CPOL and CPHA bits to determine the SPI mode;

Configure DEF to determine data word length;

Configure LSBFIRST to determine frame format;

Configure the NSS pin, such as setting the SSOE bit to let the hardware set the NSS. You can also set the SSM bit and make the SSI position high;

To set the MSTR bit and the SPE bit, you need to make sure that the NSS is already high.

When you need to transmit data, you only need to write the data to be transmitted to the data register. SPI will send the data from the send buffer to the shift register in parallel, and then send the data from the shift register as set by LSBFIRST. When the data has reached the shift register, the TXE flag will be set, and if TXEIE has been set, an interrupt will occur. If the TXE flag position bit needs to fill the data register, maintain the complete data flow.

When the receiver receives the data, when the last sampling clock edge of the data word arrives, the data is transferred from the shift register to the receive buffer in parallel, the RXNE bit is set, and an interrupt occurs if the RXNEIE bit is previously set. At this point, the data register should be read and the data should be removed as soon as possible.

Figure 16-2 SPI master mode read/write mode 0

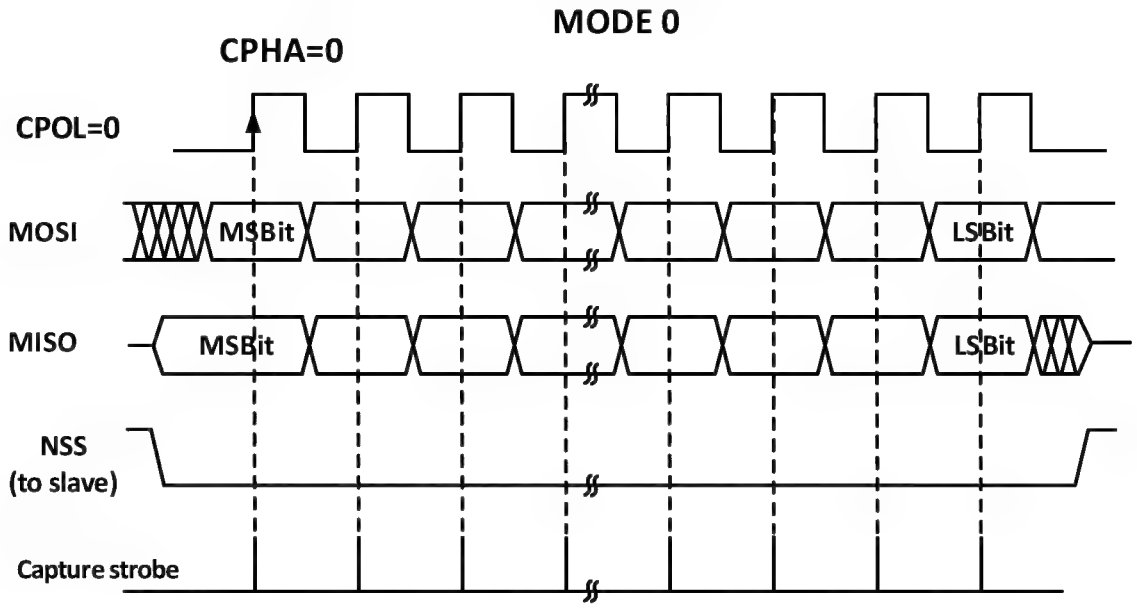


Figure 16-3 SPI master mode read/write mode 1

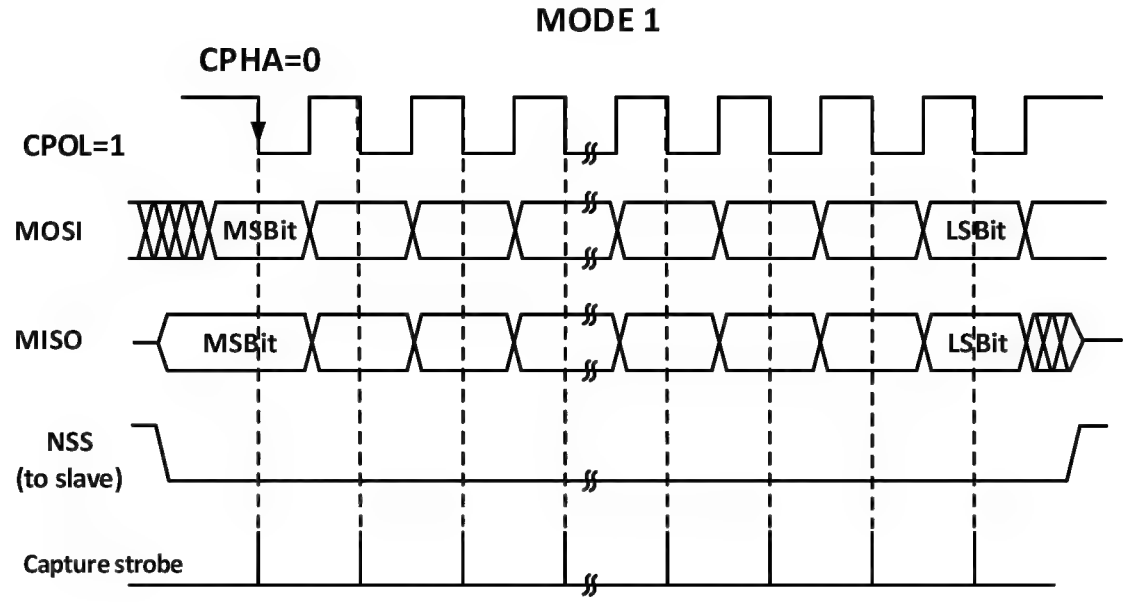


Figure 16-4 SPI master mode read/write mode 2

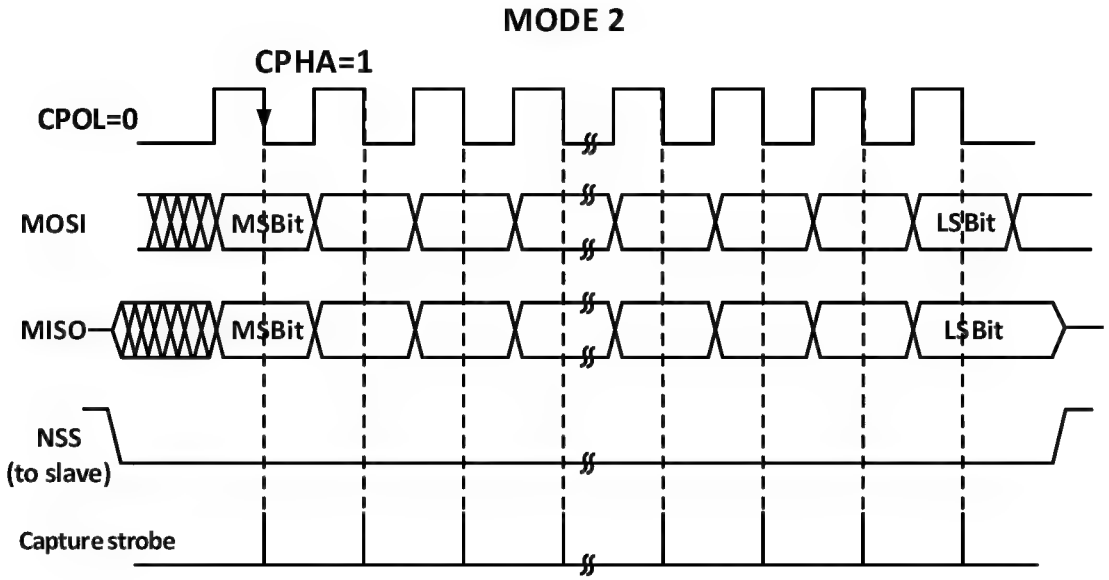
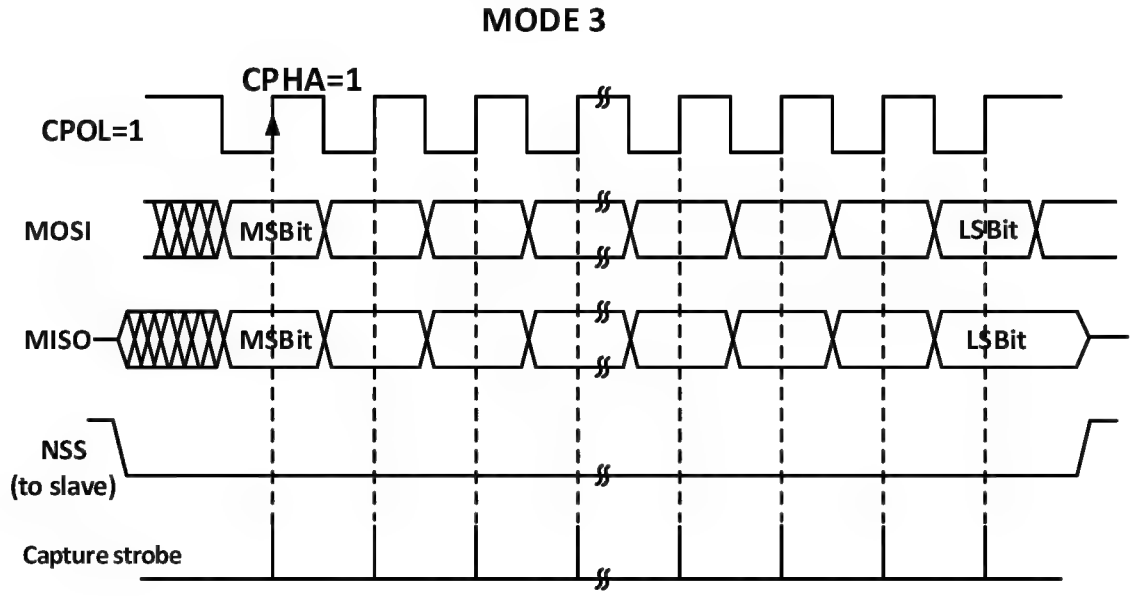


Figure 16-5 SPI master mode read/write mode 3



16.2.3 Slave Mode

When the SPI module works in slave mode, SCK is used to receive the clock sent by the host, and its own baud rate setting is invalid. The steps to configure to slave mode are as follows:

Configure the DEF bit to set the data bit length;

Configure CPHA to match the host operating mode;

The working mode of SPI is determined according to the configuration of transceiver and CPOL.

If you need to send in slave mode, you need to set CPOL to mode 2 or mode 3, and the host will change the configuration as needed.

If you only need to receive from the mode, you only need to match the host CPOL mode.

Configure LSBFIRST to match host data frame format;

In hardware management mode, the NSS pin needs to be kept low. If NSS is set to software management (SSM setting), please keep the SSI not set.

Clear MSTR bit, set SPE bit, turn on SPI mode.

At the time of transmission, when the first slave receiving sampling edge appears in the SCK, the slave starts sending. The process of sending is that the data of the send buffer is moved to the send shift register. When the data of the send buffer is moved to the shift register, the TXE flag will be set. If the TXEIE bit is set before, then an interrupt will occur.

When receiving, after the last clock sampling edge, the RXNE bit is set, the bytes received by the shift register are transferred to the receiving buffer, and the data in the receiving buffer can be obtained by reading the data register. If the RXNEIE is set before the RXNE is set, an interruption occurs.

Figure 16-6 SPI slave mode read/write mode 0

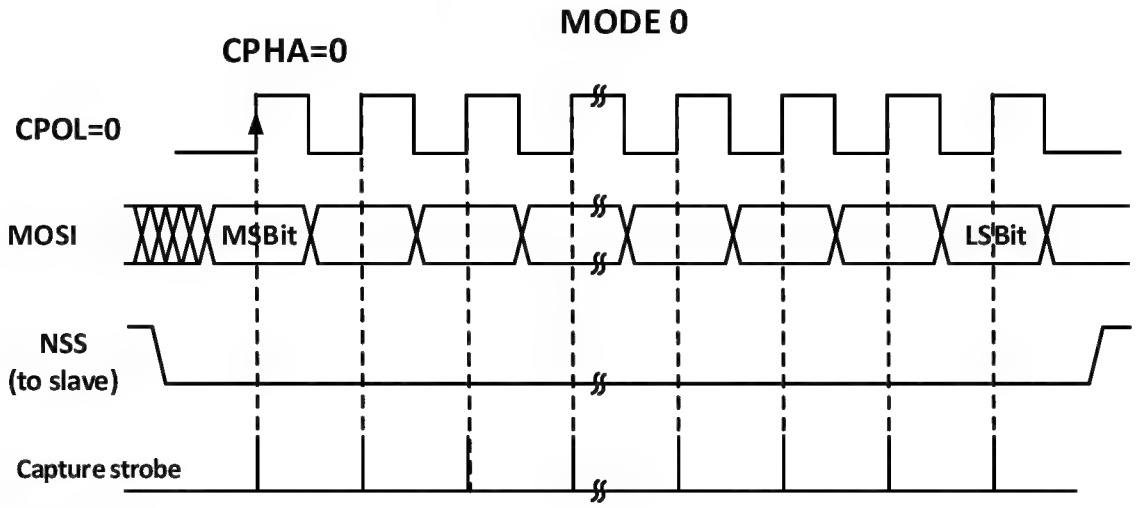


Figure 16-7 SPI slave mode read/write mode 1

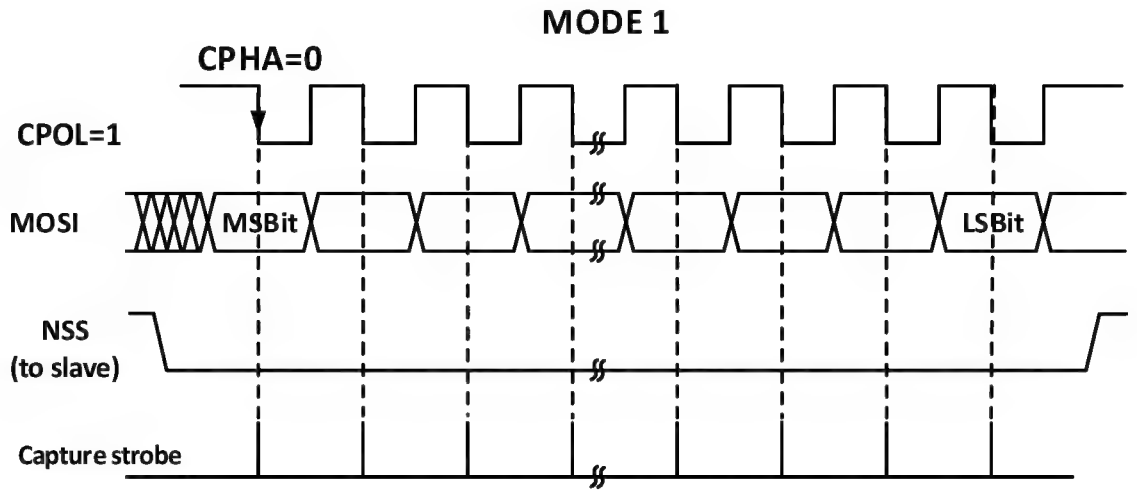


Figure 16-8 SPI slave mode read/write mode 2

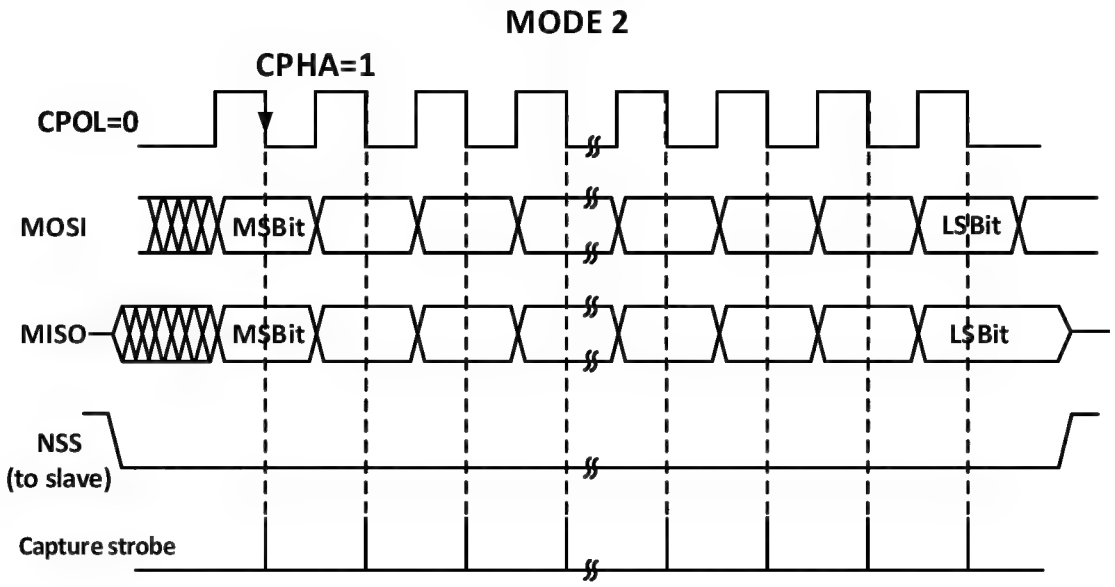
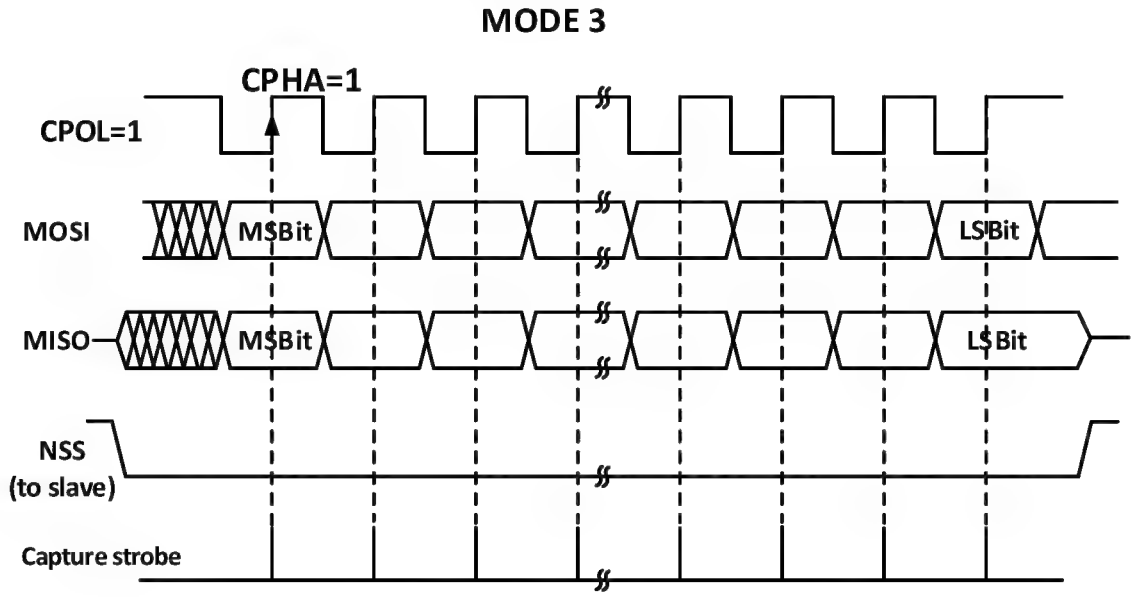


Figure 16-9 SPI slave mode read/write mode 3



16.2.4 Simplex Mode

The SPI interface can operate in half-duplex mode, where the master device uses the MOSI pin and the slave device uses the MISO pin for communication. When using half-duplex communication, you need to set BIDIMODE and use BIDIOE to control the transmission direction.

Setting the RXONLY bit in normal full-duplex mode sets the SPI module to receive-only simplex mode, releasing a data pin after RXONLY is set. The SPI can also be set to transmit only mode by ignoring the received data.

16.2.5 CRC

The SPI module uses CRC checksum to ensure the reliability of full-duplex communication, and separate CRC calculators are used for data sending and receiving. the polynomial for CRC calculation is determined by the

polynomial register, and different calculations are used for 8-bit data width and 16-bit data width, respectively. Setting the CRCEN bit will enable CRC checksum and at the same time will reset the CRC calculator. After the last data byte is sent, setting the CRCNEXT bit will send the TXCRCR calculator calculation after the current byte is sent, while the CRCERR bit will be set if the last received receive shift register value does not match the locally calculated RXCRCR calculation. Using the CRC checksum requires setting the polynomial calculator and setting the CRCEN bit when configuring the SPI operating mode, and setting the CRCNEXT bit on the last word or half-word to send the CRC and perform the receive CRC checksum. Note that the polynomial for the CRC calculation should be unified for both transmitting and receiving.

16.2.6 DMA

The SPI module supports the use of DMA to speed up data communication, either by using DMA to fill the transmit buffer or by using DMA to pick up data from the receive buffer in a timely manner. DMA will pick up or send data in a timely manner using RXNE and TXE as signals. DMA can also operate in simplex or CRC mode.

16.2.7 Errors

- Master mode fault (MODF)

When the SPI is operating in NSS pin hardware management mode, an external pull-down of the NSS pin occurs; or in NSS pin software management mode, the SSI bit is cleared; or the SPE bit is cleared, causing the SPI to be shut down; or the MSTR bit is cleared and the SPI enters slave mode. If the ERRIE bit is already set, an interrupt is also generated. Steps to clear the MODF bit: First perform a read or write operation to R16_SPI1_STATR, and then write R16_SPI1_CTLR1.

- Overrun condition

If the host transmits data and there is unread data in the receive buffer of the slave device, an overflow error occurs, the OVR bit is set, and an interrupt is also generated if ERRIE is set. Transmitting an overflow error should restart the current transmission. Reading the data register and then reading the status register will eliminate this bit.

- CRC error

When the received CRC word and the value of RXCRCR do not match, a CRC error will be generated and the CRCERR bit will be set.

16.2.8 Interrupts

The SPI module supports five interrupt sources, among which the TXE and RXNE events are set when the TXEIE and RXNEIE bits are set respectively. In addition to the above three errors will also generate interrupts, namely MODF, OVR and CRCERR, after enabling the ERRIE bit, these three errors will also generate error interrupts.

16.3 Register Description

Table 16-2 SPI-related registers list

Name	Offset address	Description	Reset value
R16_SPI1_CTLR1	0x40013000	SPI Control Register 1	0x0000
R16_SPI1_CTLR2	0x40013004	SPI Control Register 2	0x0000
R16_SPI1_STATR	0x40013008	SPI Status Register	0x0002
R16_SPI1_DATAR	0x4001300C	SPI Data Register	0x0000

R16_SPI1_CRCR	0x40013010	SPI Polynomial Register	0x0007
R16_SPI1_RCRCR	0x40013014	SPI Receive CRC Register	0x0000
R16_SPI1_TCRCR	0x40013018	SPI Transmit CRC Register	0x0000
R16_SPI1_HSCR	0x40013024	SPI High-speed Control Register	0x0000

16.3.1 SPI Control Register 1 (SPI_CTLR1)

Offset address: 0x00

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BIDIMODE	BIDIOE	CRCEN	CRCNEXT	DFF	RXONLY	SSM	SSI	LSBFIRST	SPE	BR[2:0]			MSTR	CPOL	CPHA

Bit	Name	Access	Description	Reset value
15	BIDIMODE	RW	One-way data mode enable bit. 1: Select single-wire two-way mode; 0: Select the two- wire two-way mode.	0
14	BIDIOE	RW	Single-wire output enable bit, used in conjunction with BIDIMODE. 1: Enable output, transmit only 0: Disable output, receive only.	0
13	CRCEN	RW	The hardware CRC verifies the enable bit, which can only be written when SPE is 0 and can only be used in full-duplex mode. 1: Enable CRC calculation; 0: Disable CRC calculation.	0
12	CRCNEXT	RW	After the next data transfer, the value of the CRC register is sent. This bit should be set immediately after the last data is written to the data register. 1: Transmit CRC check result; 0: Continue to transmit data from the data register.	0
11	DFF	RW	Data frame length bit, which can only be written when SPE is 0. 1: Transmit and receive using a 16-bit data length; 0: Transmit and receive using a 16-bit data length.	0
10	RXONLY	RW	Only bits are received in dual-wire mode, which is used in conjunction with BIDIMODE. Setting this bit allows the device to only receive but not send. 1: Receive only, simplex mode; 0: Full-duplex mode.	0
9	SSM	RW	Chip selection pin management bit. This bit determines whether the level of the NSS pin is controlled by hardware or software. 1: Software control NSS pin; 0: Hardware control NSS pin.	0

8	SSI	RW	Chip selection pin control bit. In the case of SSM setting, this bit determines the level of the NSS pin. 1: NSS is high; 0: NSS is low.	0
7	LSBFIRST	RW	Frame format control bits. This bit cannot be modified during communication. 1: Transmit LSB first; 0: Transmit MSB first	0
6	SPE	RW	SPI enable bit. 1: Enable SPI; 0: Disable SPI.	0
[5:3]	BR[2:0]	RW	The baud rate sets the domain, which cannot be modified during communication. 000: $F_{HCLK}/2$; 001: $F_{HCLK}/4$; 010: $F_{HCLK}/8$; 011: $F_{HCLK}/16$; 100: $F_{HCLK}/32$; 101: $F_{HCLK}/64$; 110: $F_{HCLK}/128$; 111: $F_{HCLK}/256$. *Note: This bit only applies when the HSRXEN bit is 0; when the HSRXEN bit is 1, the SCK frequency is FHCLK/ (BR+2).*	0
2	MSTR	RW	Master-slave setting bit, which cannot be modified during communication. 1: Configure as the master device; 0: Configure as the slave device.	0
1	CPOL	RW	Clock polarity selection bit, which cannot be modified during communication. 1: SCK stays high when idle; 0: SCK stays low when idle. *Note: When using SPI to send data from mode, you should set this bit to 1 and select mode 2 or mode 3.*	0
0	CPHA	RW	Clock phase setting bit, which cannot be modified during communication. 1: Data sampling starts from the second clock edge; 0: Data sampling starts from the first clock edge.	0

16.3.2 SPI Control Register 2 (SPI_CTLR2)

Offset address: 0x04

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								TXEIE	RXNEIE	ERRIE	Reserved		SSOE	TXDMAEN	RXDMAEN

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	TXEIE	RW	Send buffer air break enable bit. Setting this bit allows an interrupt to occur when TXE is set.	0
6	RXNEIE	RW	Receive buffer non-null interrupt enable bit. Setting this bit allows an interrupt to occur when RXNE is set.	0
5	ERRIE	RW	Error interrupt enable bit. Setting this bit allows an interrupt to occur when an error (CRCERR, OVR, MODF) is generated.	0
[4:3]	Reserved	RO	Reserved	0
2	SSOE	RW	SS output enable. Disable SS output can work in multi-master mode. 1: Enable SS output; 0: Disable SS output in master mode.	0
1	TXDMAEN	RW	Transmit buffer DMA enable bit. 1: Enable transmit buffer DMA; 0: Disable transmit buffer DMA.	0
0	RXDMAEN	RW	Receive buffer DMA enable bit. 1: Enable receive buffer DMA; 0: Disable receive buffer DMA.	0

16.3.3 SPI Status Register (SPI_STATR)

Offset address: 0x08

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								BSY	OVR	MODF	CRCERR	Reserved		TXE	RXNE

Bit	Name	Access	Description	Reset value
[15:8]	Reserved	RO	Reserved	0
7	BSY	RO	Busy flag bit, which is set or reset by hardware. 1: SPI is communicating, or the send buffer is not empty 0: SPI is not communicating.	0
6	OVR	RW0	Overflow flag bit, which is set by hardware and reset by software. 1: An overflow error occurred 0: No overflow error occurred	0
5	MODF	RO	Mode error flag bit, which is set by hardware and reset by software. 1: A mode error occurred; 0: No mode error occurred.	0

4	CRCERR	RW0	CRC error flag bit, which is set by hardware and reset by software. 1: The received CRC value is not consistent with the RCRCR value; 0: The received CRC value is the same as the RCRCR value.	0
[3:2]	Reserved	RO	Reserved	0
1	TXE	RO	The transmit buffer is an empty flag bit. 1: Transmit buffer is empty; 0: Transmit buffer is not empty.	1
0	RXNE	RO	Receive buffer non-empty flag bits. 1: Receive buffer is not empty; 0: Receive buffer is empty *Note: Read DATAR and automatically clear zero.*	0

16.3.4 SPI Data Register (SPI_DATAR)

Offset address: 0x0C

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	DR[15:0]	RW	Data register. The data register is used to store the received data or pre-store the data to be sent, so the read and write of the data register is actually corresponding to the different operation area, in which the read pair uses the receiving buffer and the write corresponds to the transmitting buffer. The reception and transmission of data can be 8-bit or 16-bit, and how many bits of data need to be determined before transmission. When using 8 bits for data transmission, only the lower 8 bits of the data register are used, and the high 8 bits are forced to 0 when received. The use of 16-bit data structures causes all 16-bit data registers to be used.	0

16.3.5 SPI Polynomial Register (SPI1_CRCR)

Offset address: 0x10

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRCPOLY[15:0]															

Bit	Name	Access	Description	Reset value

[15:0]	CRCPOLY[15:0]	RW	CRC polynomials. This field defines the polynomials used in CRC calculations.	7h

16.3.6 SPI Receive CRC Register (SPI1_RCRCR)

Offset address: 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRC[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	RXCRC[15:0]	RO	Receive CRC value. The calculated result of the CRC check of the received bytes is stored. Setting CRCEN resets the register. The calculation method uses the polynomials used by CRCPOLY. In 8-bit mode, only the lower 8 bits participate in the calculation, while in 16-bit mode, all 16 bits participate in the calculation. You need to read this register when BSY is 0.	0

16.3.7 SPI Transmit CRC Register (SPI1_TCRCR)

Offset address: 0x18

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXCRC[15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	TXCRC[15:0]	RO	Transmit CRC. Stores the calculated results of the CRC check of the bytes that have been sent. Setting CRCEN resets the register. The calculation method uses the polynomials used by CRCPOLY. In 8-bit mode, only the lower 8 bits participate in the calculation, while in 16-bit mode, all 16 bits participate in the calculation. You need to read this register when BSY is 0.	0

16.3.8 SPI High-speed Control Register (SPI1_HSCR)

Offset address: 0x24

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved															HSRX EN

Bit	Name	Access	Description	Reset value
[15:1]	Reserved	RO	Reserved	0

0	HSRXEN	RW	Read enable in SPI high-speed mode: 1: Enable high-speed reading mode; 0: Disable high-speed reading mode.	0

Chapter 17 Operational Amplifier (OPA) and Comparator (CMP)

This chapter OPA module description applies only to CH32V005, CH32V006, CH32V007 and CH32M007 products, where CH32V005 products do not have OPA polling functionality, and this chapter CMP module description applies only to CH32V007 and CH32M007 products.

The module consists of an independently configurable operational amplifier (OPA or PGA) and an independently configurable voltage comparator (CMP1), as well as a voltage comparator (CMP2) used in conjunction with the operational amplifier, in which the operational amplifier (OPA or PGA) supports gain selection or can be used as a voltage comparator.

The input and output of the operational amplifier can be connected to the I/O port, and the input channel (Pin) and gain are optional, the output channel can be optionally configured to the general I/O port, and an additional internal output channel is directly connected to the ADC internal channel to support the external analog small signal amplification into the ADC to achieve small signal ADC conversion. High-speed mode is supported, and the pendulum rate can be increased by setting high-speed mode.

The input and output of each voltage comparator are connected to the I/O port, and the input pins are optional, and the output pins can be optionally configured to the universal I/O port or multiplexed as a TIM internal sampling channel (Without using the I/O pin).

17.1 Main Features

- OPA input channel (Pin) selectable
- The OPA output channel can be selected from the universal I/O port, and the internal output channel is directly connected to the ADC internal channel.
- OPA supports front-end input polling.
- OPA supports single-ended PGA, differential PGA and PGA gain selection.
- CMP1 input pin selectable
- The CMP output pin can choose either the universal I/O port or the TIM internal sampling channel.
- 1 interrupt vector

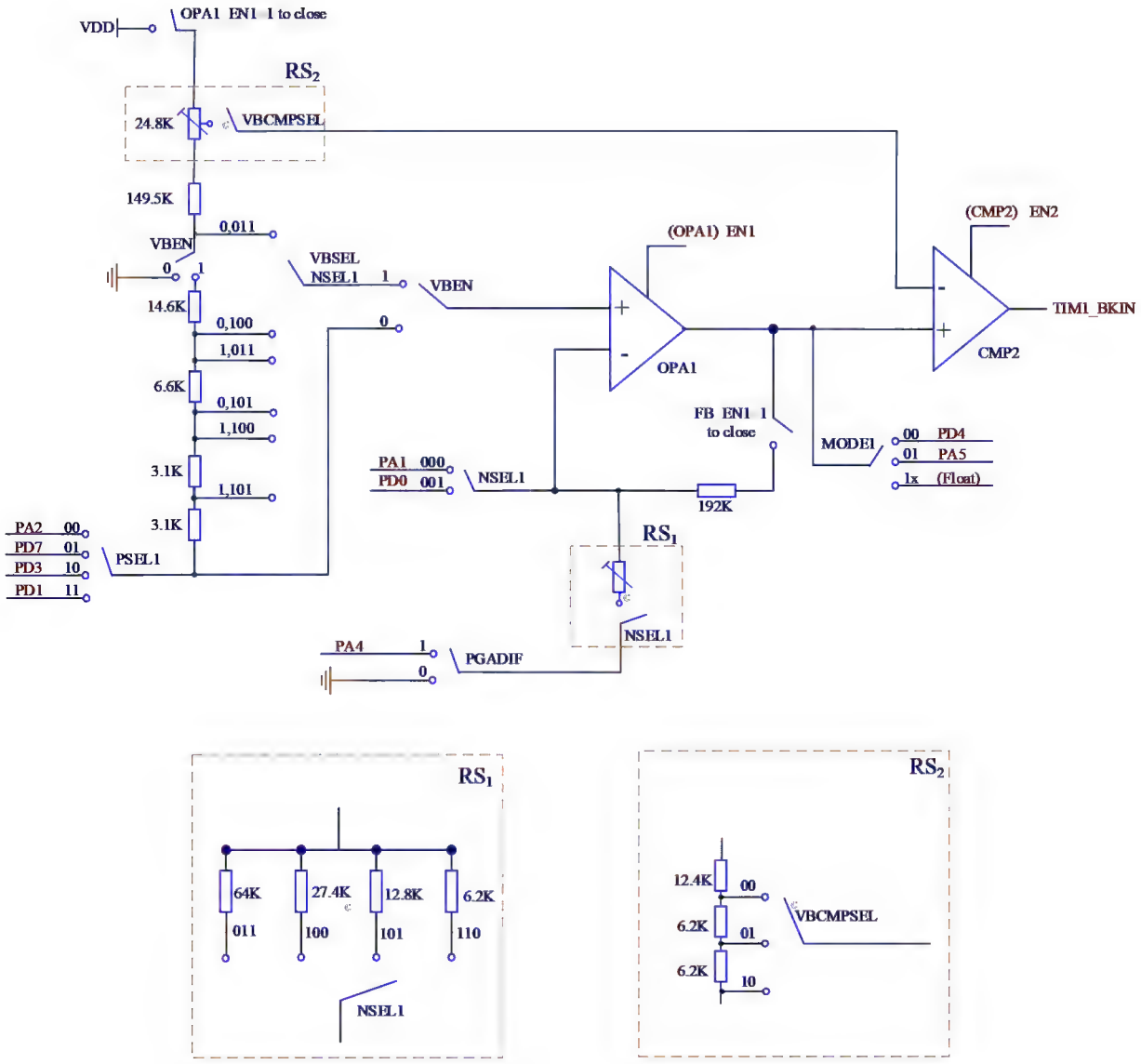
17.2 Function Description

17.2.1 OPA

In the OPA_CTLR1 register: configure the OPA_EN1 bit to enable the OPA1; to configure the MODE1 bit to select the output channel of the OPA1; configure the PSEL1 bit to select the positive input pin of the OPA1; and configure the NSEL1 bit to select the negative input channel of the OPA1 or the gain when used as the PGA.

In the OPA_CTLR1 register: configure the OPA_HS1 bit to turn on the OPA1 high-speed mode and increase the swing rate to 40V/μs; configure the PGADIF bit to turn on the differential input PGA mode; configure the VBEN bit to turn on the bias function PGA mode, and cooperate with the CMP_EN2 bit in the VBCMPSEL and OPA_CTLR2 registers to compare the OPA output with the bias voltage.

Figure 17-1 OPA structure diagram

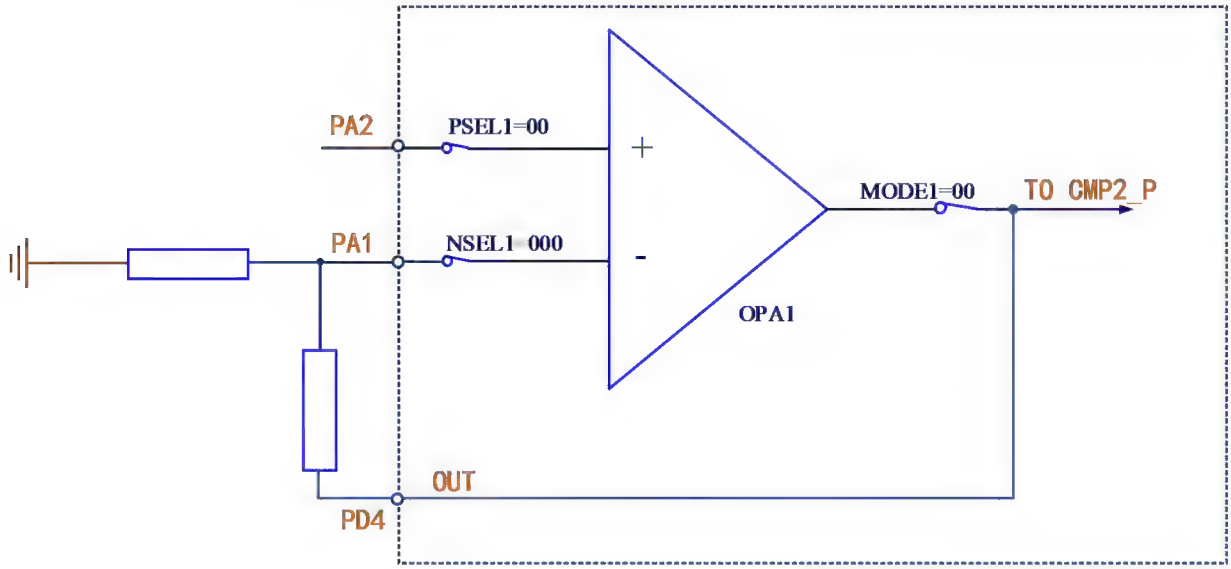


17.2.1.1 OPA Single-ended Input

In the OPA_CTLR1 register: configure PSEL1 bit to select OPA1 positive and negative pin input; configure MODE1 bit to select OPA1 positive and negative pin output, and the corresponding pin is configured as analog input. After OPA_EN1 position 1 enables OPA1, the external pin can be linked internally.

Note: External access feedback resistor can be selected when using stand-alone OPA1.

Figure 17-2 Schematic diagram of OPA single-ended input



Configuration actions:

1. Unlock OPA:

Write KEY1 = 0x45670123 to OPA_KEY register (The first step must be KEY1);

Write KEY2 = 0xCDEF89AB to OPA_KEY register (The second step must be KEY2).

2. Select OPA pin:

In the OPA_CTLR1 register: Configure the PSEL1 bit to 00b (PA2), NSEL1 bit to 000b (PA1), and MODE1 bit to 00b (PD4).

3. In the OPA_CTLR1 register: Configure the OPA_EN1 bit to 1, enable OPA1.

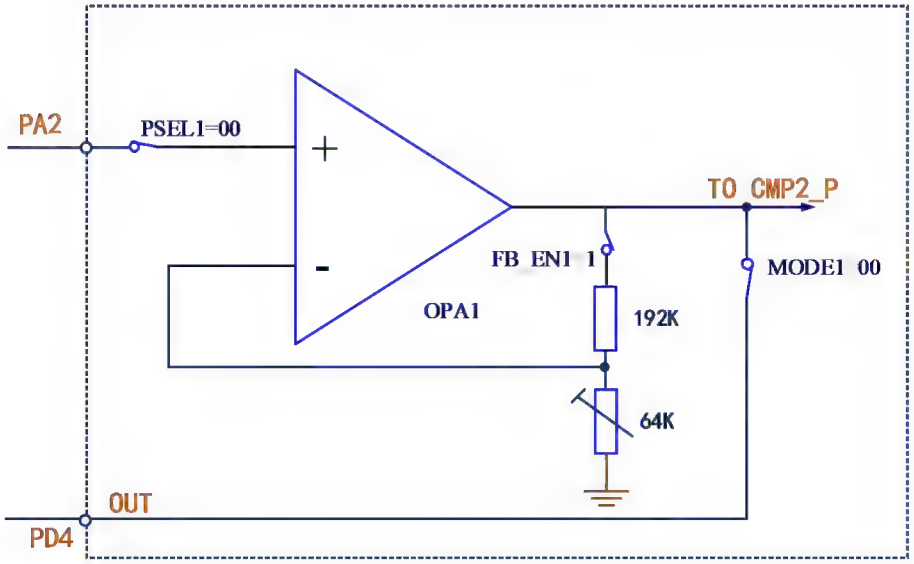
17.2.1.2 OPA Built-in Programmable PGA

In the OPA_CTLR1 register: configure the PGADIF bit to 0, select the OPA single-ended mode, so that the negative input is internally grounded; configure the NSEL1 bit to be 011bUniple 100b, 101bpx10b, 1010b, the gain is: 4pinch 816shock 32, and FB_EN1 position 1, enable the internal feedback resistance; configure the PSEL1 bit to select the positive input; configure the OPA_EN1 bit to 1, enable OPA1.

Note:

1) *In internally programmable PGA mode, the external pin only needs the input and output of the positive end of the OPA, and the negative input defaults to the internal ground.*
2) *After selecting the gain, in order to avoid output saturation, the input voltage should be lower than VDD divided by the gain multiple.*

Figure 17-3 Schematic diagram of OPA internal programmable PGA



Configuration actions:

1. Unlock OPA:

Write KEY1 = 0x45670123 to OPA_KEY register (The first step must be KEY1);

Write KEY2 = 0xCDEF89AB to OPA_KEY register (The second step must be KEY2).

2. Select OPA pin:

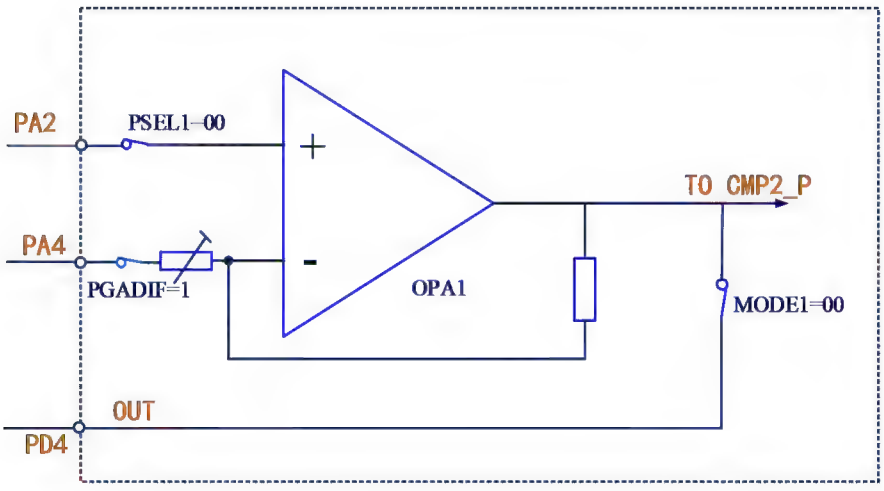
In the OPA_CTLR1 register: Configure the PSEL1 bit to 00b (PA2), NSEL1 bit to 000b (PA1), and MODE1 bit to 00b (PD4).

3. In the OPA_CTLR1 register: Configure the OPA_EN1 bit to 1, enable OPA1.

17.2.1.3 OPA Differential Input

In the OPA_CTLR1 register: configure the PGADIF bit to 1, select the OPA differential input PGA mode, then the negative input is configured with the NSEL1 bit 011b PGA, and the OPA_EN1 bit is configured as 1 to enable internal feedback resistor; the PSEL1 bit is configured to select the positive input; and the OPA_EN1 bit is configured as 1 to enable OPA1.

Figure 17-4 Schematic diagram of OPA differential input

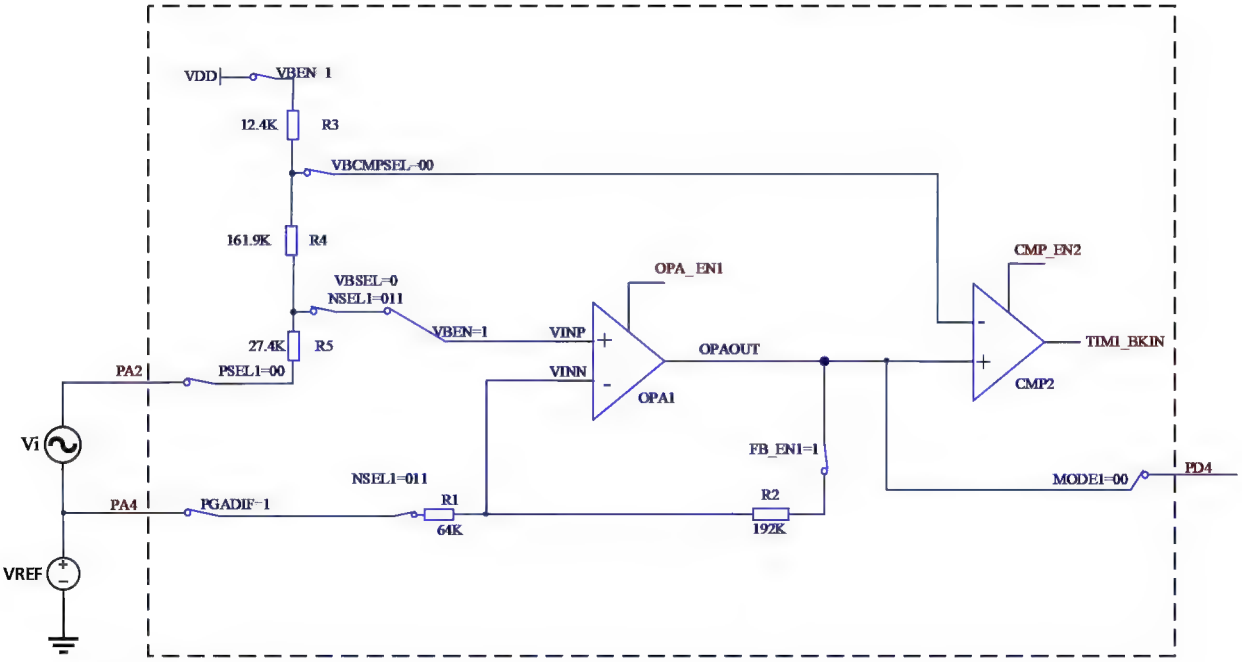


17.2.1.4 OPA Self-biased Differential PGA

When the OPA uses differential PGA, the bias voltage can be increased, and the output voltage can be connected to the positive input of the CMP2, and then compared with the negative reference voltage.

In the OPA_CFGR2 register: configure the RST_EN2 bit to 1 to enable the CMP2 reset function, when the CMP output is a high-level complex system, or configure the BKIN_CFG bit of the OPA_CTLR2 register to 10b as the brake source of the TIM1.

Figure 17-5 Schematic diagram of OPA self-offset differential PGA input



Configuration actions:

1. Unlock OPA:

Write KEY1 = 0x45670123 to OPA_KEY register (The first step must be KEY1);

Write KEY2 = 0xCDEF89AB to OPA_KEY register (The second step must be KEY2).

2. Select OPA pin:

In the OPA_CTLR1 register: Configure OPA_EN1 bit 1, enable OPA1; Configure PSEL1 bit 00b (PA2); NSEL is 011b, and FB_EN1 position 1, enable internal feedback resistance; MODE1 is 00b (PD4).

3. Configure VBEN to 1b, enable output voltage bias; VBSEL is 0b, select PGA output reference voltage.

4. CMP unlock:

Write KEY1 = 0x45670123 to CMP_KEY register (The first step must be KEY1);

Write KEY2 = 0xCDEF89AB to CMP_KEY register (The second step must be KEY2).

5. In the OPA_CTLR1 register: Configure the VBCMPSEL bit to 00b and give the reference voltage at the negative end of the CMP2.

6. In the OPA_CFGR2 register: Configure BKIN_CFG to 10b as the brake source of TIM1.

7. In the OPA_CFGR2 register: Configure the CMP_EN2 bit to 1, enable CMP2.

The calculation methods of OPA output bias voltage VB_{OPAOUT}, OPA output AC gain $Gain_{OPAOUT}$ and CMP negative terminal voltage VB_{CMP} are given according to figure 17-5, as shown in Table 17-1, Table 17-2 and Table 17-3:

Table 17-1 OPA output bias voltage VB_{OPAOUT}

VBSEL	VB_{OPAOUT}
VBSEL=0	(VDD+VREF)/2
VBSEL=1	VDD/4+VREF*3/4

Table 17-2 OPA output AC gain $Gain_{OPAOUT}$

VBSEL	$Gain_{OPAOUT}$
VBSEL=0	((R1+R2)/R1)-0.5
VBSEL=1	((R1+R2)/R1)-0.25

Table 17-3 CMP negative terminal voltage VB_{CMP}

VBCMPSEL	VB_{CMP}
VBCMPSEL=00	VDD*30/33+VINP*3/33
VBCMPSEL=01	VDD*29/33+VINP*4/33
VBCMPSEL=10	VDD*28/33+VINP*5/33

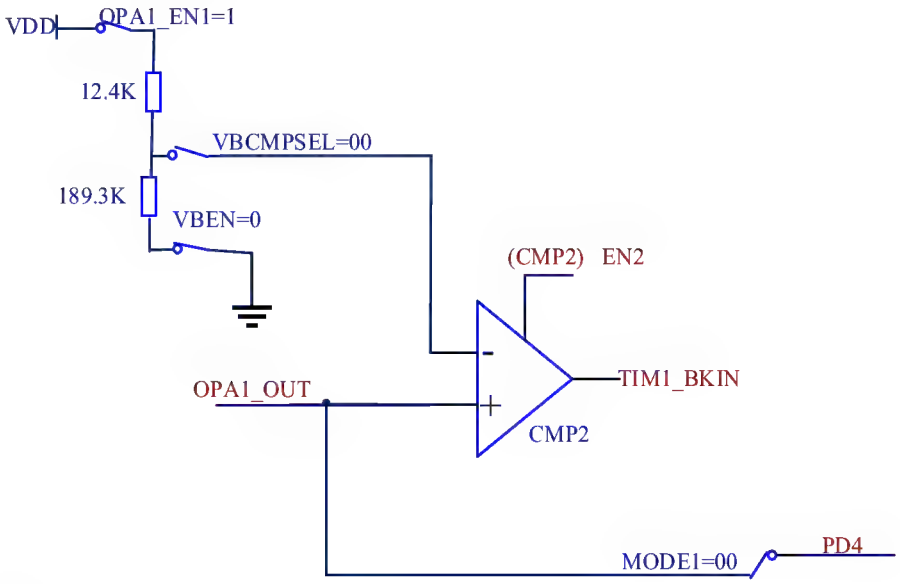
For applications where VREF is unknown, the AC gain can be obtained in similar ways as follows:

1) NSEL=011b, the ADC1 is obtained by sampling OPA_OUT of the above-mentioned 1-7 ADC.
2) NSEL=100b, repeat the sampling OPA_OUT of the above-mentioned 1-7, ADC samples OPA_OUT to get ADC2.
3) Then Vi=(ADC2-ADC1)/4.

17.2.1.5 CMP2 Independent Bias Mode

When CMP2 is used alone, the negative bias voltage of CMP2 can be increased independently, and the output can be compared with the results of OPA1_OUT. In the OPA_CFGR2 register: configure the RST_EN2 bit to 1, enable the CMP2 reset function, when the CMP output is the high-level reset system, or configure the BKIN_CFG bit of the OPA_CTLR2 register to 10b as the brake source of the TIM1.

Figure 17-6 Schematic diagram of CMP2 independent bias mode



Configuration actions:

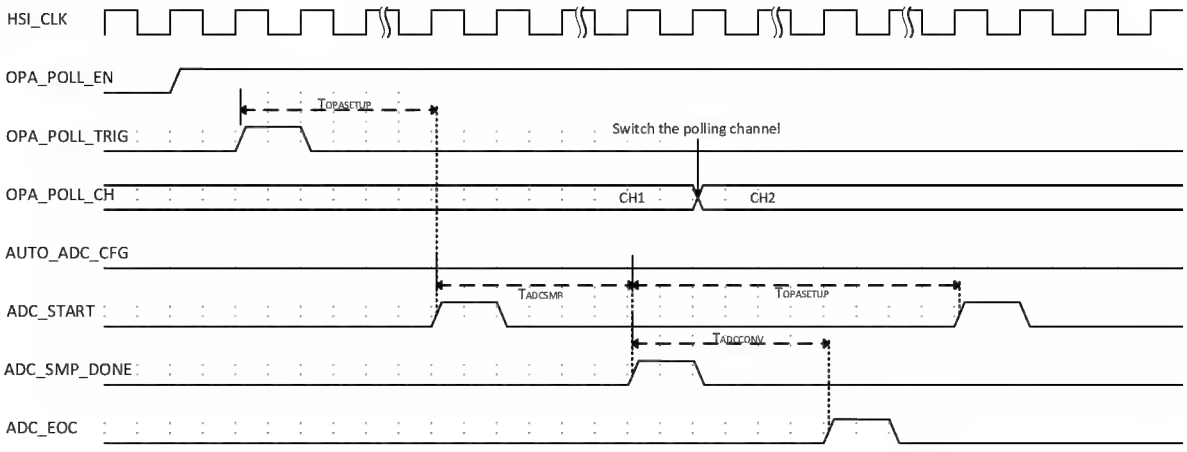
1. Unlock OPA:

Write KEY1 = 0x45670123 to OPA_KEY register (The first step must be KEY1);
Write KEY2 = 0xCDEF89AB to OPA_KEY register (The second step must be KEY2).
2. Select OPA pin:
In OPA_CTLR1 register: Configure OPA_EN1 bit to 1, enable OPA1.
3. Configure VBEN to 0b, enable output voltage bias.
4. CMP unlock:
Write KEY1 = 0x45670123 to CMP_KEY register (The first step must be KEY1);
Write KEY2 = 0xCDEF89AB to CMP_KEY register (The second step must be KEY2).
5. In the OPA_CTLR1 register: Configure the VBCMPSEL bit to 00b and give the reference voltage at the negative end of the CMP2.
6. In the OPA_CFGR2 register: Configure BKIN_CFG to 10b as the brake source of TIM1.
7. In the OPA_CFGR2 register: Configure the CMP_EN2 bit to 1, enable CMP2.

17.2.2 OPA Positive Input Polling

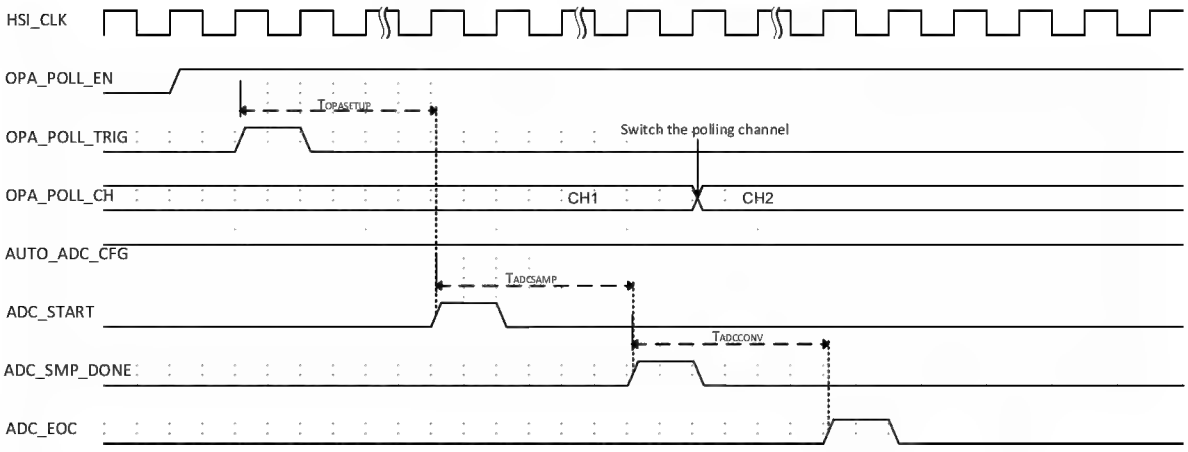
The P terminals of OPA can be selected from OPA_P0/OPA_P1/OPA_P2/OPA_P3, and the number of channels to be polled can be configured through the POLL1_NUM in the OPA_CFGR1 register, and up to three channels can be selected; the polling function of OPA can be realized by selecting the OPA_P0/OPA_P1/OPA_P2/OPA_P3 in turn. The polling function of OPA can realize to select OPA_P0/OPA_P1/OPA_P2/OPA_P3 sequentially at regular intervals to select all P terminals in turn; OPA polling order can be set by configuring the POLL_CHx bit in the OPA_CFGR1 register, and POLL_EN bit can be configured to turn on the polling of the positive end of OPA1.

Figure 17-7 OPA polling timing diagram 1



$T_{OPASETUP}$: OPA establishment time.
T_{ADCSMP}: ADC sampling time.
$T_{ADCCONV}$: ADC conversion time.
As shown in Figure 17-7, when the AUTO_ADC_CFG bit is configured to 0, after the configured polling trigger event occurs, the ADC starts to sample after TOPASETUP, after the sampling is completed, the polling channel is switched, and then after TOPASETUP again (Note: the setup OPA establishment time needs to be greater than the ADC conversion time), the next ADC sampling is performed, and the ADC sampling is repeated after the process.

Figure 17-8 OPA polling timing diagram 2



.

$T_{OPASETUP}$: OPA establishment time.

T_{ADCSMP}: ADC sampling time.

$T_{ADCCONV}$: ADC conversion time.

As shown in Figure 17-8, when the AUTO_ADC_CFG bit is configured to 1, after the configured polling trigger event occurs, the ADC starts to sample after TOPASETUP, and the polling channel is switched after the sampling is completed; wait for the subsequent polling trigger event to occur again and repeat the above process.

Recommendation: In practice, if the ADC conversion time is greater than the OPA establishment time, it is recommended that the AUTO_ADC_CFG bit be configured to 1 to work with the ADC scanning mode; if the ADC conversion time is less than the OPA establishment time, it is recommended that the AUTO_ADC_CFG bit be configured to 0 to work with the ADC one-shot mode.

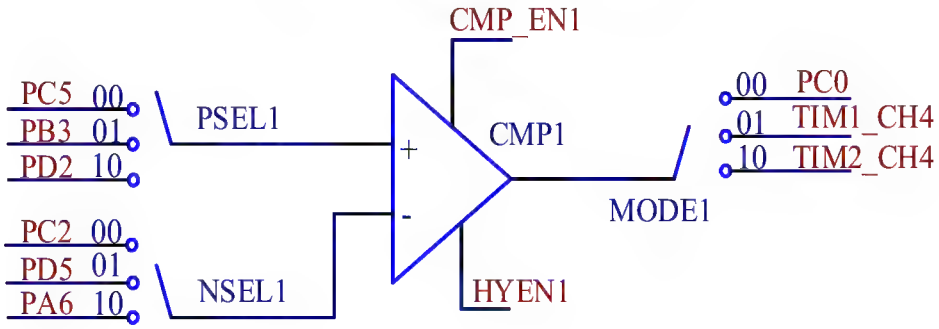
17.2.3 CMP

Setting the CMP_ENx in the OPA_CTLR2 register enables the corresponding CMPx.

In the OPA_CTLR2 register: configure MODE1 bit to select the output channel of CMP1; configure PSEL1 to select the positive input pin of CMP1; configure NSEL1 to select the negative input pin of CMP1.

The CMP2 positive input pin comes from the OPA1 output and can select the channel through the MODE1 bit in the OPA_CTLR1 register; the negative reference voltage is controlled by the VBCMPSEL bit in the OPA_CTLR register and is valid only in VBEN=1. The output of CMP2 is used as the brake source of TIM1.

Figure 17-9 CMP structure diagram



17.3 Register Description

Table 17-4 OPA-related registers list

Name	Offset address	Description	Reset value
R32_OPA_CFGR1	0x40024000	OPA Configuration Register 1	0x00000000
R32_OPA_CTLR1	0x40024004	OPA Control Register 1	0x800C0736
R32_OPA_CFGR2	0x40024008	OPA Configuration Register 2	0x00000000
R32_OPA_CTLR2	0x4002400C	OPA Control Register 2	0x80000078
R32_OPA_KEY	0x40024010	OPA Lock Key Register	0xXXXXXXXX
R32_CMP_KEY	0x40024014	CMP Unlock Key Register	0xXXXXXXXX
R32_POLL_KEY	0x40024018	POLL Lock Key Register	0xXXXXXXXX

17.3.1 OPA Configuration Register 1 (OPA_CFGR1)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POLL_LOCK	Reserved			POLL_SEL			POLL_SWSTRT	Reserved		POLL_CH3		POLL_CH2		POLL_CH1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	IF_OUT_POLL_CH3	IF_OUT_POLL_CH2	IF_OUT_POLL_CH1	Reserved	NMI_EN	Reserved	IE_OUT1	AUTO_ADC_CFG	SETUP_CFG		RST_EN1	POLL1_NUM		Reserved	POLL_EN

Bit	Name	Access	Description	Reset value
31	POLL_LOCK	RO	POLL lock: 1: POLL lock; 0: POLL unlock. *Note: The module is not reset and this bit cannot be unlocked.*	0
[30:28]	Reserved	RO	Reserved	0
[27:25]	POLL_SEL	RW	OPA polling trigger event selection: 000: Software configuration; 001: TIM1_CH4; 010: TIM2_CH4; 011: TIM3_CH1; 100: TIM3_CH2; Other: Reserved.	0
24	POLL_SWSTRT	WO	To start OPA polling, you need to set a software trigger. 1: Start OPA polling; 0: Reset status. This bit is set by software, and the hardware is cleared 0 after polling starts.	0

[23:22]	Reserved	RO	Reserved	0
[21:20]	POLL_CH3	RW	OPA polling order: Configure the 3rd polling channel.	0
[19:18]	POLL_CH2	RW	OPA polling order: Configure the 2nd polling channel.	0
[17:16]	POLL_CH1	RW	OPA polling order: Configure the 1st polling channel.	0
15	Reserved	RO	Reserved	0
14	IF_OUT_POLL_CH3	RW0	The 3rd polling channel polls the OPA1 for high output interrupt flags: 1: OPA output is high; 0: Invalid.	0
13	IF_OUT_POLL_CH2	RW0	The 2nd polling channel polls the OPA1 for high output interrupt flags: 1: OPA output is high; 0: Invalid.	0
12	IF_OUT_POLL_CH1	RW0	The 1std polling channel polls the OPA1 for high output interrupt flags: 1: OPA output is high; 0: Invalid.	0
11	Reserved	RO	Reserved	0
10	NMI_EN	RW	OPA connection NMI interrupt enable: 1: Enable; 0: Disable.	0
9	Reserved	RO	Reserved	0
8	IE_OUT1	RW	OPA1 interrupt enable: 1: Turn on interrupt enable; 0: Turn off interrupt enable.	0
7	AUTO_ADC_CFG	RW	OPA polling automatically triggers ADC transformation configuration bits: 1: Under the polling function, OPA switches channels after ADC sampling ends; 0: Under the polling function, after the ADC sampling ends, the OPA switches the channel, and the ADC conversion is automatically triggered after the SETUP_CFG setting time.	0
[6:5]	SETUP_CFG	RW	OPA establishes the time configuration. After establishing the time, the ADC conversion will be triggered: 00, 10: 0.5μs; 01: 0.312μs; 11: 0.77μs.	0
4	RST_EN1	RW	OPA1 reset system enable: 1: Enable the OPA1 reset function. The result of OPA1 polling is that the system will be reset at high power level.	0

			0: Disable the OPA1 reset function	
[3:2]	POLL1_NUM	RW	Configure the number of positive ends of OPA1 polling: 00: Poll 1 channel; 01: Poll 2 channel; 10: Poll 3 channel; 11: Reserved. *Note: The polling order is configurable.*	0
1	Reserved	RO	Reserved	0
0	POLL_EN	RW	OPA1 front-end polling enable: 1: Turn on the OPA1 front-end polling enable; 0: Turn off the OPA1 front-end polling enable.	0

17.3.2 OPA Control Register 1 (OPA_CTLR1)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA_LOCK	Reserved										OPA_HS1	VBCMPSEL		VBSEL	VBEN

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			PGADIF	FB_EN1	NSEL1			Reserved		PSEL1		Reserved	MODE1		OPA_EN1

Bit	Name	Access	Description	Reset value
31	OPA_LOCK	RW1	OPA lock (Write 1 lock, write 0 invalid): 1: OPA lock; 0: OPA unlock.	1
[30:21]	Reserved	RO	Reserved	0
20	OPA_HS1	RW	OPA1 high-speed mode enable: 1: Enable, increase the pendulum rate to 40V/μs; 0: Disable.	0
[19:18]	VBCMPSEL	RW	Given the reference voltage at the negative end of CMP2, only VBEN=1 is valid: 00, 01, 10: See Table 17-3 for details; 11: Turn off.	11b
17	VBSEL	RW	PGA mode, positive reference voltage selection: 1: VDD/4; 0: VDD/2.	0
16	VBEN	RW	PGA mode, positive reference voltage enable: 1: Enable; 0: Disable.	0
[15:13]	Reserved	RO	Reserved	0
12	PGADIF	RW	OPA is used as a PGA with NSEL1, and the N-terminal is connected to OPA_CHN2 (PA4). 1: Turn on differential input PGA mode;	0

			0: Turn off differential input PGA mode.	
11	FB_EN1	RW	OPA1 internal feedback resistor enable: 1: Enable, used as PGA1, feedback resistance is about 192 kΩ; 0: Disable. *Note: When NSEL1 control bit is in PGA mode, this bit must be set to 1.*	0
[10:8]	NSEL1	RW	OPA1 negative input channel and gain selection when used as PGA: 000: OPA_CHN0 (PA1); 001: OPA_CHN1 (PD0); 011: PGA mode, no negative input channel, internal gain 4; 100: PGA mode, no negative input channel, internal gain 8; 101: PGA mode, no negative input channel, internal gain 16; 110: PGA mode, no negative input channel, internal gain 32; 010, 111: Off.	111b
[7:6]	Reserved	RO	Reserved	0
[5:4]	PSEL1	RW	Input channel selection at the front end (P side) of OPA1: 00: OPA_CHP0 (PA2); 01: OPA_CHP1 (PD7); 10: OPA_CHP2 (PD3); 11: OPA_CHP3 (PD1).	11b
3	Reserved	RO	Reserved	0
[2:1]	MODE1	RW	OPA1 output channel selection: 00: PD4, internal CMP2 positive input; 01: PA5, internal CMP2 positive input; 1x: The internal CMP2 is on the input.	11b
0	OPA_EN1	RW	OPA1 enable: 1: Enable; 0: Disable.	0

17.3.3 OPA Configuration Register 2 (OPA_CFGR2)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	POLL_CH3		POLL_CH2		POLL_CH1		POLL_VLU								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF_CNT	IF_OUT_POL_CH3	IF_OUT_POL_CH2	IF_OUT_POL_CH1	Reserved		IE_CNT	IE_OUT1	Reserved		RST_EN2	_RST_EN1	POLL_NUM		Reserved	POLL_EN1

Bit	Name	Access	Description	Reset value
31	Reserved	RO	Reserved	0
[30:29]	POLL_CH3	RW	CMP1 polling sequence configuration: Configure the 3rd polling channel.	0
[28:27]	POLL_CH2	RW	CMP1 polling sequence configuration: Configure the 2nd polling channel.	0
[26:25]	POLL_CH1	RW	CMP1 polling sequence configuration: Configure the 1st polling channel.	0
[24:16]	POLL_VLU	RW	Configure CMP1 front-end polling interval: Polling interval = (POLL_VLU+1)*1μs。	0
15	IF_CNT	RW0	CMP1 polling interval end flag: 1: End of polling interval 0: Invalid. Write 0 clear zero, write 1 is invalid.	0
14	IF_OUT_POLL_CH3	RW0	The 3rd polling channel polls the flag that the CMP1 output is high: 1: Polled to CMP1 output is high; 0: Invalid. Write 0 clear zero, write 1 is invalid.	0
13	IF_OUT_POLL_CH2	RW0	The 2nd polling channel polls the flag that the CMP1 output is high: 1: Polled to CMP1 output is high; 0: Invalid. Write 0 clear zero, write 1 is invalid.	0
12	IF_OUT_POLL_CH1	RW0	The 1st polling channel polls the flag that the CMP1 output is high: 1: Polled to CMP1 output is high; 0: Invalid. Write 0 clear zero, write 1 is invalid.	0
[11:10]	Reserved	RO	Reserved	0
9	IE_CNT	RW	Interrupt enabling at the end of the CMP1 polling interval: 1: Enable; 0: Disable.	0
8	IE_OUT1	RW	CMP1 interrupt enable: 1: Enable; 0: Disable.	0
[7:6]	Reserved	RO	Reserved	0
5	RST_EN2	RW	CMP2 reset system enable: 1: Enable CMP2 reset function, CMP1 output to high level will reset the system; 0: Disable.	0
4	RST_EN1	RW	CMP1 reset system enable: 1: Enable CMP1 reset function, CMP1 output to high level will reset the system;	0

			0: Disable.	
[3:2]	POLL1_NUM	RW	Configure the number of positive ends of CMP1 polling: 00: 1 channel; 01: 2 channels; 10: 3 channels; 11: Reserved. *Note:The polling order is configurable.*	0
1	Reserved	RO	Reserved	0
0	POLL_EN1	RW	CMP1 front-end polling enable: 1: Enable; 0: Disable.	0

17.3.4 OPA Control Register 2 (OPA_CTLR2)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP_LOCK	Reserved			BKIN_CFG		FILT_SEL	FILT_EN	Reserved							CMP_EN2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved							RMID1	HYEN1	PSEL1		NSEL1		MODE1		CMP_EN1

Bit	Name	Access	Description	Reset value
31	CMP_LOCK	RW1	CMP lock (Write 1 lock, write 0 invalid): 1: Lock CMP; 0: Unlock CMP.	1
[30:28]	Reserved	RO	Reserved	0
[27:26]	BKIN_CFG	RW	TIM1 brake source configuration: 00: IO; 01: CMP1; 10: CMP2; 11: OPA.	0
25	FILT_SEL	RW	CMP output digital filter length selection: 1: 0.5μs; 0: 0.33μs.	0
24	FILT_EN	RW	CMP digital filter enable: 1: Enable; 0: Disable.	0
[23:17]	Reserved	RO	Reserved	0
16	CMP_EN2	RW	CMP2 enable: 1: Enable; 0: Disable.	0
[15:9]	Reserved	RO	Reserved	0
8	RMID1	RW	CMP1 positive input channel virtual center point enable:	0

			1: Enable; 0: Disable.	
7	HYEN1	RW	CMP1 as hysteresis comparator switch: 1: On, ±24mV; 0: Off.	0
[6:5]	PSEL1	RW	CMP1 front-end input channel selection: 00: COMP_CHP0 (PC5); 01: COMP_CHP1 (PB3); 10: COMP_CHP2 (PD2); 11: Invalid.	11b
[4:3]	NSEL1	RW	CMP1 negative input channel selection: 00: COMP_CHN0 (PC2); 01: COMP_CHN1 (PD5); 10: COMP_CHN2 (PA6); 11: Invalid.	11b
[2:1]	MODE1	RW	CMP1 output mode selection: 00: CMP1 result output from PC0; 01: CMP1 result internal directly connected to TIM1_CH4; 10: CMP1 result internal directly connected to TIM2_CH4; 11: Reserved.	0
0	CMP_EN1	RW	CMP1 enable: 1: Enable; 0: Disable.	0

17.3.5 OPA Unlock Key Register 2 (OPA_KEY)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPA_KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPA_KEY[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	OPA_KEY[31:0]	W0	OPA key, the locking keys for input OPA include: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

17.3.6 CMP Unlock Key Register (CMP_KEY)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CMP_KEY[31:16]															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP_KEY[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	CMP_KEY[31:0]	W0	CMP key, the locking keys for input CMP include: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

17.3.7 POLL Lock Key Register (POLL_KEY)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POLL_KEY[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLL_KEY[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	POLL_KEY[31:0]	W0	Poll key, the locking keys for input POLL include: KEY1 = 0x45670123; KEY2 = 0xCDEF89AB. After locking, it can be unlocked only if the system is reset.	X

Chapter 18 Flash Memory and User Option Bytes (FLASH)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

18.1 Flash Memory Organization

The internal flash memory of the chip is organized as follows (Take CH32V006 as an example).

Table 18-1 Flash Memory Organization

Block	Name	Address Range	Size (Byte)
Main memory	Page 0	0x0800 0000 – 0x0800 00FF	256
	Page 1	0x0800 0100 – 0x0800 01FF	256
	Page 2	0x0800 0200 – 0x0800 02FF	256
	Page 3	0x0800 0300 – 0x0800 03FF	256
	…	…	…
	Page 247	0x0800 F700– 0x0800 F7FF	256
Information block	Startup program code	0x1FFF 0000 – 0x1FFF 0CFF	3K+256
	User option bytes	0x1FFF F800 – 0x1FFF F8FF	256

Note: The above main memory area is used for the user's application storage and is divided into write protection units in 2K bytes (8 pages).

18.2 Flash Memory Programming and Security

18.2.1 Programming/Erasing Methods

- Fast programming: This method uses page operation (recommended). After a specific sequence of unlocking, it performs a single 256-byte programming and 256-byte erasing, 1K-byte erasing (standard 1K whole chip erasing is also applicable to fast programming).

16.2.2 Security - Prevent Illegal Access (Read, Write, Erase)

- Page write protection
- Read protection

When the chip is in the read-protected state.

1) Main memory pages 0-7 (2K bytes) are automatically write-protected state, not controlled by FLASH_WPR register; unread-protected state, all main memory pages are controlled by FLASH_WPR register.
2) System boot code area, SWD (SDI) mode, RAM area cannot erase or program the main memory, except the whole chip erase. The user option byte area can be erased or programmed. If you try to unprotect the read (programming user words), the chip will automatically erase the entire user area.

Note: The internal RC oscillator (HSI) must be turned on when programming / erasing flash memory.

18.3 Register Description

Table 18-2 FLASH-related registers list

Name	Offset address	Description	Reset value
R32_FLASH_ACTLR	0x40022000	Control Register	0x00000000

R32_FLASH_KEYR	0x40022004	FPEC Key Register	0xXXXXXXXX
R32_FLASH_OBKEYR	0x40022008	OBKEY Register	0xXXXXXXXX
R32_FLASH_STATR	0x4002200C	Status Register	0x0000B000
R32_FLASH_CTLR	0x40022010	Configuration Register	0x0000X080
R32_FLASH_ADDR	0x40022014	Address Register	0xXXXXXXXX
R32_FLASH_OBR	0x4002201C	Option Byte Register	0x0XXXXXXX
R32_FLASH_WPR	0x40022020	Write Protection Register	0xXXXXXXXX
R32_FLASH_MODEKEYR	0x40022024	Extended Key Register	0xXXXXXXXX
R32_FLASH_BOOT_MODEKEYR	0x40022028	Unlock BOOT Key Register	0xXXXXXXXX

18.3.1 Control Register (FLASH_ACTLR)

Offset address: 0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														LATENCY	

Bit	Name	Access	Description	Reset value
[31:2]	Reserved	RO	Reserved	0
[1:0]	LATENCY[1:0]	RW	FLASH wait status number. 00: 0 wait (Recommended 0<=SYSCLK<=15MHz); 01: 1 wait (Recommended 15<SYSCLK<=24MHz); 10: 2 wait (Recommended 24<SYSCLK<=48MHz); 11: Reserved.	0

18.3.2 FPEC Key Register (FLASH_KEYR)

Offset address: 0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	KEYR[31:0]	WO	FPEC key, the unlock keys for entering FPEC include: RDPRT key = 0x000000A5; KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

18.3.3 OBKEY Register (FLASH_OBKEYR)

Offset address: 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OBKEYR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OBKEYR[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	OBKEYR[31:0]	WO	Option byte key, which is used to enter a select word key to release OBWRE. KEY1 = 0x45670123; KEY2 = 0xCDEF89AB. (Note: FLASH needs to be unlocked first)	X

18.3.4 Status Register (FLASH_STATR)

Offset address: 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_LOCK	BOOT_MODE	BOOT_STATUS	BOOT_AVA	Reserved				TURBO	FWAKE_FLAG	EOP	WRPRTERR	Reserved			BSY

Bit	Name	Access	Description	Reset value
[31:16]	Reserved	RO	Reserved	0
15	BOOT_LOCK	RW	BOOT lock. 1: Locked, unable to write to FLASH_STATA [14] field; 0: Unlock, you can write to the FLASH_STATR [14] field. *Note: Write 1 setting, write 0 is invalid.*	1
14	BOOT_MODE	RW	Combined with BOOT_AVA, the switch between user area and BOOT area can be controlled. 1: After software reset, you can switch to the BOOT area. 0: After software reset, you can switch to the user area.	0
13	BOOT_STATUS	RO	The source of the currently executed program. 1: Indicates a program loaded from the BOOT area; 0: indicates a program loaded from the user area.	1
12	BOOT_AVA	RO	Initializes the configuration word status. 1: Indicates booting from the BOOT area; 0: Use of the BOOT area is not supported.	1
[11:8]	Reserved	RO	Reserved	0
7	TURBO	RO	TURBO mode enable. 1: Indicates that you are in TURBO mode;	0

			0: No effect.	
6	FWAKE_FLAG	RW0	FLASH wakeup flag, write 0 clear. 1: FALSH is waken up; 0: No effect.	0
5	EOP	RW1	Indicates the end of the operation, write 1 clear. The hardware is set each time it is successfully erased or programmed.	0
4	WRPRTERR	RW1	Indicates write protection error, write 1 clear. The hardware will set the address if it is programmed for write protection.	0
[3:1]	Reserved	RO	Reserved	0
0	BSY	RO	Indicates busy status. 1: Indicates that a flash operation is in progress. 0: End of operation.	0

Note: When performing the programming operation, you need to make sure the STRT bit of FLASH_CTLR register is 0.

18.3.5 Configuration Register (FLASH_ CTLR)

Offset address: 0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								BER32	Reserved			BUFRST	BUFLOAD	FTER	FTPG

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLOCK	Reserved	FWAKEIE	EOPIE	Reserved	ERRIE	OBWRE	Reserved	LOCK	STRT	OBER	Reserved		MER	PER	Reserved

Bit	Name	Access	Description	Reset value
[31:24]	Reserved	RO	Reserved	0
23	BER32	RW	Perform block erasure 32KB. *Note: Erase only addresses 0x08000000-0x08007FFF.*	0
[22:20]	Reserved	RO	Reserved	0
19	BUFRST	RW	Buffer BUF reset operation.	0
18	BUFLOAD	RW	Cache the data into BUF.	0
17	FTER	RW	Perform a fast page (256Byte) erase operation.	0
16	FTPG	RW	Perform a quick page programming operation.	0
15	FLOCK	RW1	Fast programming lock. You can only write'1'. When the bit is'1', quick programming / erase mode is not available. After the correct unlock sequence is detected, the hardware clears this bit as'0'. The software is set to 1 and re-locked.	1
14	Reserved	RO	Reserved	0
13	FWAKEIE	RW	Wake up interrupt enable, when FLASH wakes up from low-power mode, it will generate a flag	

			bit, if wake up interrupt enable is set, it will generate an interrupt signal, otherwise it will have no effect.	
12	EOPIE	RW	The operation completes the interrupt control (EOP setting in FLASH_STATR register). 1: Interrupts are allowed; 0: Interrupts is prohibited.	0
11	Reserved	RO	Reserved	0
10	ERRIE	RW	Error status interrupt control (PGERR/WRPRTERR set in FLASH_STATR register). 1: Interrupts are allowed; 0: Interrupts is prohibited.	0
9	OBWRE	RW0	User option byte lock, software clears 0. 1: Indicates that the word selected by the user can be programmed. The correct sequence needs to be written in the FLASH_OBKEYR register and then set by the hardware. 0: The user option bytes are re-locked after the software is cleared.	0
8	Reserved	RO	Reserved	0
7	LOCK	RW1	Lock. You can only write'1'. When the bit is'1', it means that FPEC and FLASH_CTLR are locked unwritable. After the correct unlock sequence is detected, the hardware clears this bit as'0'. After an unsuccessful unlock operation, the bit will not change until the next time the system is reset.	1
6	STRT	RW1	Start. Setting 1 initiates an erase action, and the hardware clears 0 automatically (BSY becomes '0').	0
5	OBER	RW	Perform user option byte erasure.	0
[4:3]	Reserved	RO	Reserved	0
2	MER	RW	Perform a full erase operation (erase the entire user area).	0
1	PER	RW	Sector erasure (1K) is performed.	0
0	Reserved	RO	Reserved	0

18.3.6 Address Register (FLASH_ ADDR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FAR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

FAR[15:0]

Bit	Name	Access	Description	Reset value
[31:0]	FAR[31:0]	WO	Flash address, which is the programming address for programming and the starting address for erasure. The register cannot be written when the BSY bit in the FLASH_STATR register is '1'.	X

18.3.7 Option Byte Register (FLASH_OBR)

Offset address: 0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved						DATA1								DATA0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA0						FIX_11		STATR_MODE	RST_MODE		STANDY_RST	Reserved	IWDG_SW	RDPRT	OBERR

Bit	Name		Access	Description	Reset value
[31:26]	Reserved		RO	Reserved	0
[25:18]	DATA1[7:0]		RO	Data byte 1.	X
[17:10]	DATA0[7:0]		RO	Data byte 0.	X
[9:8]	FIX_11		RO	Fixed at 11.	11b
7	USER	STATR_MODE	RO	Power-on startup mode: 1: Startup from BOOT area; 0: Startup from user area.	1
[6:5]		RST_MODE	RO	Configure byte reset delay time.	X
4		STANDY_RST	RO	The system is reset and controlled in standby mode, active low.	X
3		Reserved	RO	Reserved	X
2		IWDG_SW	RO	Independent watchdog (IWDG) hardware enable bit, active low.	1
1	RDPRT		RO	Read protection status. 1: Indicates that the current read protection of flash memory is valid.	1
0	OBERR		RO	Option byte error. 1: Indicates that the option byte does not match its inverse code.	0

18.3.8 Write Protect Register (FLASH_WPR)

Offset address: 0x20

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WPR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WPR[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	WPR[31:0]	RO	Flash write protection status. 1: Write protection invalid; 0: Write protection valid. Each bit represents 2K bytes (8 pages) to store write-protected status.	X

Note: The WPR is loaded from the user option byte area after the system is reset.

18.3.9 Extended Key Register (FLASH_MODEKEYR)

Offset address: 0x24

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODEKEYR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODEKEYR[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	MODEKEYR[31:0]	WO	Enter the following sequence to unlock quick programming / erase mode. KEY1 = 0x45670123; KEY2 = 0xCDEF89AB. (Note: FLASH needs to be unlocked first)	X

18.3.10 BOOT Key Register (FLASH_BOOT_MODEKEYP)

Offset address: 0x28

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODEKEYR[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODEKEYR[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	MODEKEYR[31:0]	WO	Enter the following sequence to unlock the BOOT area. KEY1 = 0x45670123; KEY2 = 0xCDEF89AB.	X

18.4 Flash Memory Operation Flow

18.4.1 Read Operations

With direct addressing in the general address space, any read operation of 8/16/32-bit data can access the contents of the flash module and get the corresponding data.

18.4.2 Unlock the Flash Memory

After a system reset, the flash controller (FPEC) and FLASH_CTLR registers are locked and inaccessible. The flash controller module can be unlocked by writing a sequence to the FLASH_KEYR register.

Unlock sequence.

1) Write KEY1 = 0x45670123 to the FLASH_KEYR register (step 1 must be KEY1).
2) Write KEY2 = 0xCDEF89AB to FLASH_KEYR register (step 2 must be KEY2).

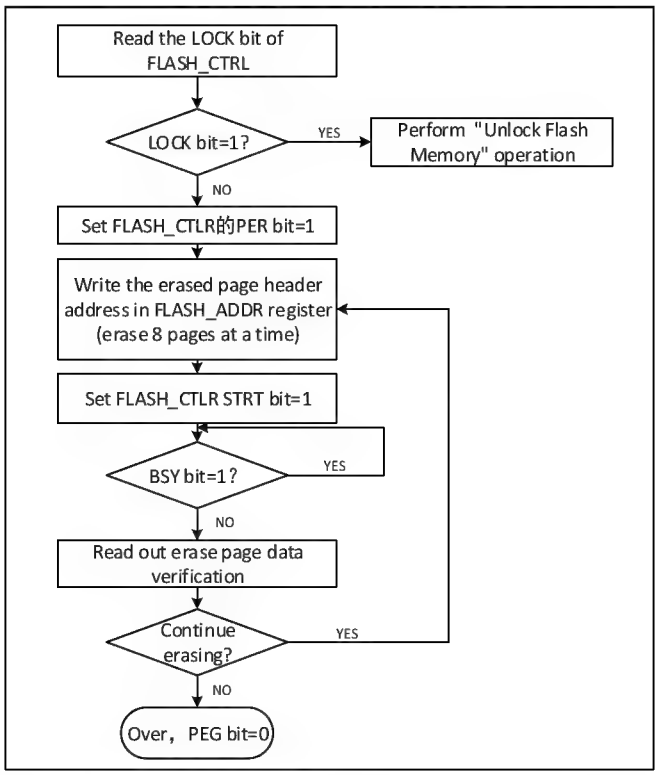
The above operations must be executed sequentially and consecutively, otherwise they are error operations and will lock the FPEC module and FLASH_CTLR registers and generate bus errors until the next system reset.

The flash memory controller (FPEC) and FLASH_CTLR registers can be locked again by setting the "LOCK" bit of the FLASH_CTLR register to 1.

18.4.3 Main Memory Standard Erase

Flash memory can be erased by standard page (1K bytes) or by whole chip.

Figure 18-1 FLASH Page Erase

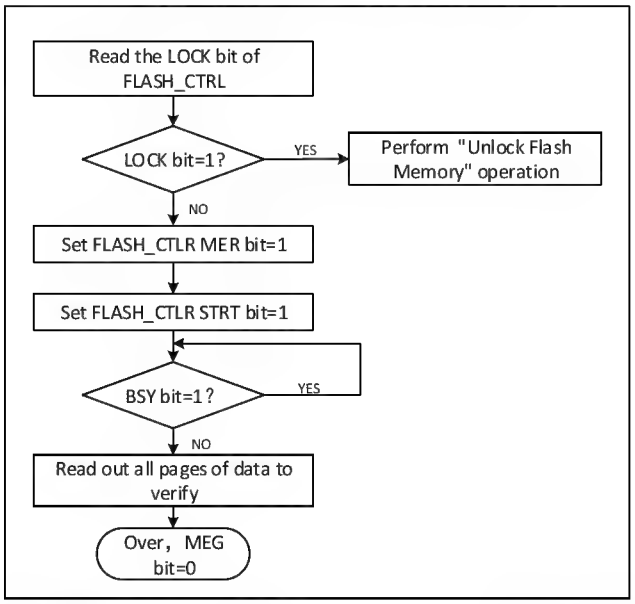


1) Check the FLASH_CTLR register LOCK bit, if it is 1, you need to perform the "unlock flash" operation.
2) Set the per bit of the FLASH_CTLR register to '1percent, and turn on the standard page erase mode.
3) Write the header address of the selected erasure to the FLASH_ADDR register.
4) Set the STRT bit of the FLASH_CTLR register to '1percent, and start an erase operation.

5) Waiting for the BSY bit to change to'0' or the EOP bit of the FLASH_STATR register to be'1' means the erasure is over, and the EOP bit is cleared to 0.
6) Read the data of the erasure page for verification.
7) Continue the standard page erasure by repeating the 3-5 steps to end the erase to clear the PEG bit 0.

Note: After erasing successfully, read the word-0xFF.

Figure 18-2 FLASH whole chip erase



1) Check the FLASH_CTLR register LOCK bit, if it is 1, you need to perform the "unlock flash" operation.
2) Set the MER bit of the FLASH_CTLR register to '1', and turn on the whole chip erase mode.
3) Set the STRT bit of the FLASH_CTLR register to '1' to start the erase action.
4) Waiting for the BSY bit to change to '0' or the EOP bit of the FLASH_STATR register to be '1' means the erasure is over, and the EOP bit is cleared to 0.
5) Read the data of the erasure page for verification.
6) Clear the MER bit 0.

18.4.4 Fast Programming Mode Unlocking

Fast programming mode operation can be unlocked by writing a sequence to the FLASH_MODEKEYR register. After unlocking, the FLOCK bit of FLASH_CTLR register will be cleared to 0, indicating that fast erase and programming operations can be performed. The FLASH_CTLR register is locked again by software setting the "FLOCK" bit to 1.

Unlock sequence.

1) Write KEY1 = 0x45670123 to the FLASH_MODEKEYR register.
2) Write KEY2 = 0xCDEF89AB to FLASH_MODEKEYR register.

The above operations must be performed sequentially and consecutively, otherwise they are wrong operations will be locked and cannot be unlocked again until the next system reset.

Note: Fast programming operation requires unlocking the "LOCK" and "FLOCK" layers.

18.4.5 Main Memory Fast Programming

Fast programming by page (256 bytes).

1) Check the LOCK bit of FLASH_CTLR register, if it is '1', you need to execute the "Unlock Flash" operation.
2) Check the FLOCK bit of FLASH_CTLR register. If it is '1', you need to perform a "Fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTPG bit of FLASH_CTLR register to enable the fast programming mode function.
5) Set the BUFRST bit of FLASH_CTLR register to perform the operation of clearing the internal 256-byte buffer.
6) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of clearing, and clear the EOP bit to 0.
7) Write data to a FLASH address in a 32-bit manner, such as
   * (uint32_t*) 0x8000000 = 0x12345678;
8) Then set the BUFLOAD bit of the FLASH_CTLR register and perform the load into the cache.
9) Wait for the BSY of the FLASH_STATR register to be '0', and write the next data.
10) Repeat steps 7-9 a total of 64 times to load all 256 bytes of data into the buffer (64 rounds of operation addresses should be consecutive).
11) Write the first address of the fast programming page to the FLASH_ADDR register.
12) Set the STRT bit of FLASH_CTLR register to '1' to start a fast page programming action.
13) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of programming, and clear the EOP bit to 0.
14) Query FLASH_STATR register to see if there is an error, or read the programmed address data checksum.
15) Continue the Quick Page programming can repeat steps 5-14 and end the programming to clear the FTPG bit to 0.

18.4.6 Main Memory Fast Erase

Fast Erase erases by page (256 bytes).

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to execute the "Fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.
4) Set the FTER bit of FLASH_CTLR register to '1' to enable the fast page erase (256 bytes) mode function.
5) Write the first address of the fast erase page to the FLASH_ADDR register.
6) Set the STRT bit of FLASH_CTLR register to '1' to initiate a fast page erase (256 bytes) action.
7) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0.
8) Query FLASH_STATR register to see if there is an error, or read the erase page address data checksum.
9) Continue fast page erase can repeat steps 5-8, end erase will FTER bit clear 0.

Note: After erasing successfully, read the word-0xFF.

Fast Erase erases by page (32 bytes).

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the FLASH_CTLR register FLOCK bit, if it is 1, you need to execute the "Fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to confirm that there are no other programming operations in progress.

4) Set the FTER bit of FLASH_CTLR register to '1' to enable the fast page erase (32 bytes) mode function.
5) Write the first address of the fast erase page to the FLASH_ADDR register.
6) Set the STRT bit of FLASH_CTLR register to '1' to initiate a fast page erase (32 bytes) action.
7) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0.
8) Query FLASH_STATR register to see if there is an error, or read the erase page address data checksum.
9) Continue fast page erase can repeat steps 5-8, end erase will BER32 bit clear 0.

Note: After erasing successfully, read the word-0xFF.

18.5 User Option Bytes

The user option bytes are fixed in the FLASH and will be reloaded into the corresponding register after the system is reset, and the user can erase and program at will. The user option bytes block with a total of 8 bytes (4 bytes for write protection, 1 byte for read protection, 1 byte for configuration options, and 2 bytes for storing user data), and each bit has its own inverse code bit for checking during loading. The information structure and meaning of the option bytes are described below.

Table 18-3 32-bit option byte format division

[31:24]	[23:16]	[15:8]	[7:0]
Option bytes byte 1 inverse code	Option bytes byte 1	Option bytes byte 0 inverse code	Option bytes byte 0

Table 18-4 User option byte information structure

Address / Bit	[31:24]	[23:16]	[15:8]	[7:0]
0x1FFFF800	nUSER	USER	nRDPR	RDPR
0x1FFFF804	nData1	Data1	nData0	Data0
0x1FFFF808	nWRPR1	WRPR1	nWRPR0	WRPR0
0x1FFFF80C	nWRPR3	WRPR3	nWRPR2	WRPR2

Name/Byte			Description	Reset value
RDPR			Read protection control bit and configure whether the code in the flash memory can be read. 0xA5: If this byte is 0xA5 (nRDP must be 0x5A), it indicates that the current code is in an unread-protected state and can be read; Other values: Indicates the code read protection status, unreadable, 0-7 pages (2K) will be automatically write-protected, not controlled by WRPR0.	0xA5
USER	[7:6]	Reserved	Reserved	11b
	5	START_MODE	Power-on startup mode: 1: Boot from BOOT area 0: Boot from user area	1

	[4:3]	RST_MODE [1:0]	Alternate as external pin reset. 00: Ignoring pin states within 128us after turning on the alternate function. 01: Ignoring pin states within 1ms after turning on the alternate function. 10: Ignoring pin states within 12ms after turning on the alternate function. 11: Alternate function off, PD7 for I/O function.	CH32V007/ CH32M007 chip reset value: 11b; Other chip reset value: 10b
	2	STANDYRST	System reset control in Standby mode: 1: Not enabled, does not reset when entering Standby mode system; 0: Enabled, generates a system reset when entering Standby mode.	1
	1	Reserved	Reserved	1
	0	IWDG_SW	Independent watchdog (IWDG) hardware enable configuration. 1: IWDG is opened by software, but hardware is not allowed; 0: The IWDG is turned on by the hardware itself (since the clock of the IWDG is provided by the LSI, the LSI will turn on automatically). *Note: If the core stops in debug mode, the watchdog hardware will fail.*	1
Data0–Data1			Stores 2 bytes of user data.	0xFFFF
WRPR0 - WRPR3			Write protection control bit. Each bit is used to control the write protection status of 2 sectors (1K bytes / sector) in the main memory: 1: Disable write protection 0: Enable write protection. 4 bytes are used to protect a total of 65K bytes of main memory. WRPO: Sector 0-15 storage write protection control; WRP1: Sector 16-31 storage write protection control; WRP2: Sector 32-47 storage write protection control; WRP3: Bit 0-6 provides write protection for sector 48-61; bit 7 provides write protection for sector 62 (3328 bytes of system memory).	0xFFFFFFFF

18.5.1 User Option Bytes Unlocking

The user option bytes operation can be unlocked by writing a sequence to the FLASH_OBKEYR register. After unlocking, the OBWRE bit of FLASH_CTLR register will be set to 1, indicating that the user option bytes can be erased and programmed. It can be locked again by clearing the "OBWRE" bit of FLASH_CTLR register to 0 by software.

Unlock sequence.

1) Write KEY1 = 0x45670123 to FLASH_OBKEYR register.
2) Write KEY2 = 0xCDEF89AB to FLASH_OBKEYR register.

Note: User-option bytes operation requires unlocking the "LOCK" and "OBWRE" layers.

18.5.2 User Option Bytes Programming

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the FLASH_CTLR register FLOCK bit. If it is '1', you need to perform a "Fast programming mode unlock" operation.
3) Check the BSY bit of the FLASH_STATR register to verify that there are no other programming operations in progress.
4) Set the FTPG bit of the FLASH_CTLR register to '1blank to enable fast page programming mode.
5) Set the BUFRST bit of the FLASH_CTLR register and perform the operation of clearing the internal 256byte cache.
6) Waiting for the BSY bit to change to'0' or the EOP bit of the FLASH_STATR register to be'1' means that the cleanup is over, and the EOP bit is cleared to 0.
7) Write data to a FLASH address in a 32-bit manner, such as *(uint32_t*)0x1FFFF804= 0x5AA55AA5;
8) Then set the BUFLOAD bit of the FLASH_CTLR register and perform the load into the cache.
9) Wait for the BSY of the FLASH_STATR register to be '0percent, and write the next data.
10) Repeat steps 7-9 for 4 times to load 16 bytes of data into the cache (the addresses of the main 16 rounds of operations should be continuous). The current page remaining bytes use the buffer reset default.
11) Write the first address of the quick programming page to the FLASH_ADDR register.
12) Set the STRT bit of the FLASH_CTLR register to '1' to start fast page programming.
13) Waiting for the BSY bit to change to'0' or the EOP bit of the FLASH_STATR register to'1' means that a quick page programming is completed and the EOP bit is cleared to 0.
14) Query the FLASH_STATR register to see if there are errors, or read the programming address data check.
15) To continue fast page programming, you can repeat steps 5-14 to end programming to clear the FTPG bit 0.

Note: When "read protected" in the modified selection word is changed to "unprotected" state, an entire erase main storage area operation is automatically performed. If you modify a selection other than read Protection, the whole erase operation will not occur.

18.5.3 User Option Bytes Erasure

Directly erase the entire 256-byte user option bytes area.

1) Check the LOCK bit of FLASH_CTLR register, if it is 1, you need to execute the "Unlock Flash" operation.
2) Check the BSY bit of the FLASH_STATR register to confirm that there is no programming operation in progress.
3) Check the OBWRE bit of FLASH_CTLR register, if it is 0, it is necessary to execute the operation of "user option bytes unlock".
4) Set the OBER bit of FLASH_CTLR register to '1', after that set the STAT bit of FLASH_CTLR register to '1' to enable the user option bytes erase.
5) Wait for the BSY bit to become '0' or the EOP bit of FLASH_STATR register to be '1' to indicate the end of erase, and clear the EOP bit to 0
6) Read and erase the address data checksum.
7) End to clear the OBER bit to 0.

18.5.4 Remove Read Protection

Whether the flash memory is read-protected or not is determined by the word selected by the user. Read the FLASH_OBR register. When the RDPRT bit is'1', it indicates that the current flash memory is in read-protected state, and the flash operation is protected by a series of read-protected states. The process of unprotecting read is as follows:

1) Erase the entire user option byte area, read the protection field RDPR, and read protection is still in effect.
2) The user chooses the word to program and writes the correct RDPR code 0xA5 to remove the read protection of the flash memory. (This step will first cause the system to automatically erase the entire piece of flash memory.)
3) A power-on reset is performed to reload the option bytes (including the new RDPR code), and the read protection is removed.

18.5.4 Remove Write Protection

Whether the flash memory is write-protected or not is determined by the user option bytes. Read the FLASH_WPR register, each bit represents 2K bytes of flash space, when the bit is'1' for non-write-protected state, for'0' for write-protected. The process of unprotecting a write is as follows:

1) Erases the entire user option bytes area;
2) Write the correct RDPR code 0xA5 to allow read access;
3) The system is reset, the option bytes are reloaded (including the new WRPR [3:0] bytes), and the write protection is removed.

Chapter 19 Electronic Signature (ESIG)

The module described in this chapter is suitable for the full range of CH32V00X microcontrollers.

The electronic signature contains chip identification information: flash area capacity and unique identity. It is burned by the manufacturer to the system storage area of the memory module when it leaves the factory and can be read through SWD (SDI) or application code.

19.1 Function Description

Flash area capacity: indicates the size that the current chip user application can use.

Unique identity: 96-bit binary code, which is unique to any microcontroller and can only be read and accessed by the user. This unique identification information can be used as the security password of the microcontroller (product), add and decrypt key, product serial number, etc., to improve the system security mechanism or identify information. All of the above can be read and accessed according to the 8-16-32 bit.

19.2 Register Description

Table 19-1 ESIG-related registers list

Name	Offset address	Description	Reset value
R16_ESIG_FLACAP	0x1FFFF7E0	Flash Capacity Register	0xXXXX
R32_ESIG_UNIID1	0x1FFFF7E8	UID Register 1	0xXXXXXXXX
R32_ESIG_UNIID2	0x1FFFF7EC	UID Register 2	0xXXXXXXXX
R32_ESIG_UNIID3	0x1FFFF7F0	UID Register 3	0xXXXXXXXX

19.2.1 Flash Capacity Register (ESIG_FLACAP)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_SIZE [15:0]															

Bit	Name	Access	Description	Reset value
[15:0]	F_SIZE[15:0]	RO	Flash capacity in Kbyte. Example: 0x0080 = 128 K bytes	X

19.2.2 UID Register (ESIG_UNIID1)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[31:16]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[15:0]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[31:0]	RO	The 0-31 bits of UID.	X

19.2.3 UID Register (ESIG_UNIID2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[63:48]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[47:32]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[63:32]	RO	The 32-63 bits of UID.	X

19.2.4 UID Register (ESIG_UNIID3)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID[95:80]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID[79:64]															

Bit	Name	Access	Description	Reset value
[31:0]	U_ID[95:64]	RO	The 64-95 digits of UID.	X

Chapter 20 Extended Configuration

20.1 Extended Configuration

The system provides an EXTEN extension configuration unit (EXTEN_CTR register). The unit uses a HB clock and performs a reset action only when the system is reset. It mainly includes the following extended control bit functions:

1) Lock-up function monitoring: If the LKUPEN field is enabled, the Lock-up situation monitoring of the system will be turned on. In case of Lock-up, the system will reset the LKUPRST field by software and set the LKUPRST field to 1. After reading, you can write 1 to clear this flag.
2) DMA reuse of TIM2: The updated DMA channel can be shared between the DMA request of TIM2 channel 4 and the updated DMA request by configuring the TIM2_DMA_REMAP bit.

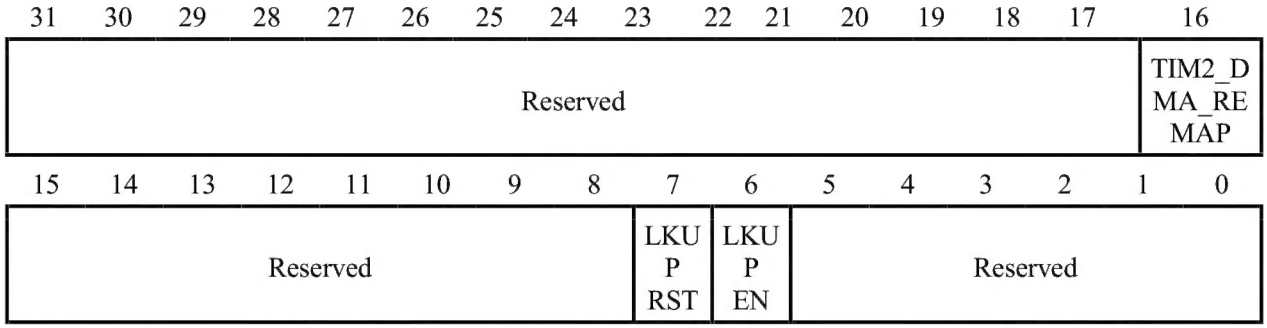
20.2 Register Description

Table 20-1 EXTEND-related registers list

Name	Access address	Description	Reset value
R32_EXTEN_CTR	0x40023800	Configure Extended Control Register	0x00000400

20.2.1 Configure Extended Control Register (EXTEND_CTR)

Offset address: 0x00



Bit	Name	Access	Description	Reset value
[31:17]	Reserved	RO	Reserved	0
16	TIM2_DMA_REMAP	RW	DMA alternate of TIM2: 1: DMA request for TIM2 Channel 4 and updated DMA Please share updated DMA channel; 0: No effect.	0
[15:8]	Reserved	RO	Reserved	0
7	LKUPRST	RW1	LOCKUP reset flag: 1: LOCKUP caused the system to reset, write 1 clear 0: Normal.	0
6	LKUPEN	RW	LOCKUP monitor function:	1

			1: Enable, perform a reset and set the LOCKUP_RST when lock-up occurs in the system; 0: Not enable.	
[5:0]	Reserved	RO	Reserved	0

Chapter 21 Debug Support (DBG)

21.1 Main Features

This register allows the MCU to be configured in the debug state. It includes:

- Independent Watchdog (IWDG) enabled counters
- Window Watchdog (WWDG) enabled counters
- Timer1 enabled counters
- Timer2 enabled counters

21.2 Register Description

21.2.1 Debug MCU Configuration Register (DBGMCU_CR)

Address: 0x7C0 (CSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	TIM3_STOP	TIM2_STOP	TIM1_STOP	Reserved		WWDG_STOP	IWDG_STOP	Reserved					STANDBY	Reserved	SLEEP

Bit	Name	Access	Description	Reset value
[31:15]	Reserved	RW	Reserved	0
14	TIM3_STOP	RW	Timer 3 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 3's counter stops working. 0: Timer 3's counter is still working normally.	0
13	TIM2_STOP	RW	Timer 2 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 2's counter stops working. 0: Timer 2's counter is still working normally.	0
12	TIM1_STOP	RW	Timer 1 debug stop bit. The counter stops when the core enters the debug state. 1: Timer 1's counter stops working. 0: Timer 1's counter is still working normally.	0
[11: 10]	Reserved	RW	Reserved	0
9	WWDG_STOP	RW	WWDG debug stop bit. The debug WWDG stops working when the core enters the debug state. 1: WWDG counter stops working. 0: WWDG counter is still working normally.	0
8	IWDG_STOP	RW	IWDG debug stop bit. The debug IWDG stops working when the core enters the debug state.	0

			1: IWDG counter stops working. 0: IWDG counter is still working normally.	
[7:3]	Reserved	RW	Reserved	0
2	STANDBY	RW	Debug standby mode bit. 1: (HCLK on) The digital circuitry section is not powered down, and the FCLK and HCLK clocks are clocked by the internal RL oscillator. Alternatively, the microcontroller exits STANDBY mode and reset by generating a system reset is the same. 0: (HCLK off) The entire digital circuitry section is powered down. From the software point of view, exiting STANDBY mode is the same as a reset (Except that some status bits indicate that the microcontroller has just exited from STANDBY state).	0
1	Reserved	RW	Reserved	0
0	SLEEP	RW	Debug sleep mode bit. 1: (HCLK on) In Sleep mode, both FCLK and HCLK clocks are provided by the originally configured system clock. 0: (HCLK off) In Sleep mode, FCLK is provided by the originally configured system clock, and HCLK is off. Since Sleep mode does not reset the configured clock system, the software does not need to reconfigure the clock system when exiting from sleep mode.	0